GE Fanuc Automation

Controllori Logici Programmabili

VersaMax PLC

Manuale Utente

GFK-1503-IT *Ottobre 1998*

Avvisi, Avvertimenti e Note Usati in questa Pubblicazione

Avvertimento

In questa pubblicazione, gli avvertimenti identificano tensioni, correnti, temperature o altre condizioni rischiose presenti nel sistema o associate al suo utilizzo che potrebbero causare lesioni personali.

Le situazioni in cui una disattenzione potrebbe causare lesioni personali o danni all’apparecchiatura sono segnalate con avvertimenti

Attenzione

Gli avvisi di attenzione vengono usati quando serve particolare cura per non danneggiare il sistema.

Nota

Le note sottolineano le informazioni particolarmente significative per comprendere il funzionamento del sistema.

Questo documento si basa sulle informazioni disponibili al momento della pubblicazione. Nonostante gli sforzi effettuati per essere precisi, le informazioni qui contenute potrebbero non rispettare tutti i dettagli o le variazioni riguardanti hardware e software e nemmeno possono prevedere tutte le situazioni che si potrebbero manifestare nell’installazione, funzionamento e manutenzione. Si potrebbero trovare descritte caratteristiche non presenti in tutti gli hardware e software. La GE Fanuc Automation non assume alcun obbligo verso chi utilizza questo documento circa informazioni riguardanti modifiche introdotte successivamente.

La GE Fanuc Automation non assume garanzie, espresse, implicite o statutarie al riguardo e non assume responsabilità per l’accuratezza, la completezza, la sufficienza o l’utilità delle informazioni qui contenute. Non vale alcuna garanzia di commerciabilità o di idoneità per gli scopi.

I seguenti sono marchi di fabbrica della GE Fanuc Automation North America, Inc.

Alarm Master
CIMPLICITY
CIMPLICITY Control
CIMPLICITY 90–ADS
CIMPLICITY PowerTRAC
Field Control
Genet
Genius
Genius PowerTRAC
Helpmate
Logicmaster
Modelmaster
Motion Mate
PowerMotion
ProLoop
PROMACRO
Series Five
Series 90
Series One
Series Six
Series Three
VuMaster
VersaMax
Workmaster

Contenuto

Contenuto

Capitolo 1 Introduzione

Guida al Manuale VersaMax

Questo manuale comprende informazioni generali sul funzionamento della CPU e il contenuto dei programmi, inoltre fornisce descrizioni dettagliate dei requisiti specifici di programmazione.

I Moduli CPU sono descritti dettagliatamente nel Capitolo 2.

Le procedure di installazione della CPU sono descritte nel capitolo 3, nel quale viene anche spiegato come rimuovere e sostituire le batterie, nonché il cablaggio della porta seriale.

La Configurazione è descritta nel capitolo 4. La configurazione determina certe caratteristiche di funzionamento dei moduli, oltre a stabilire i riferimenti del programma utilizzati da ciascun modulo del sistema.

I rimanenti capitoli descrivono dettagliatamente il funzionamento della CPU e le caratteristiche di programmazione.

- CPU Sweep: capitolo 5
- Elementi di un programma applicativo: capitolo 6
- Dati di programmazione: capitolo 7
- Funzione Service Request: capitolo 8
- I/O/SNP/RTU Seriali: capitolo 9
- La funzione PID: capitolo 10

Le appendici di questo manuale contengono dettagliate informazioni di riferimento.

Altri Manuali VersaMax

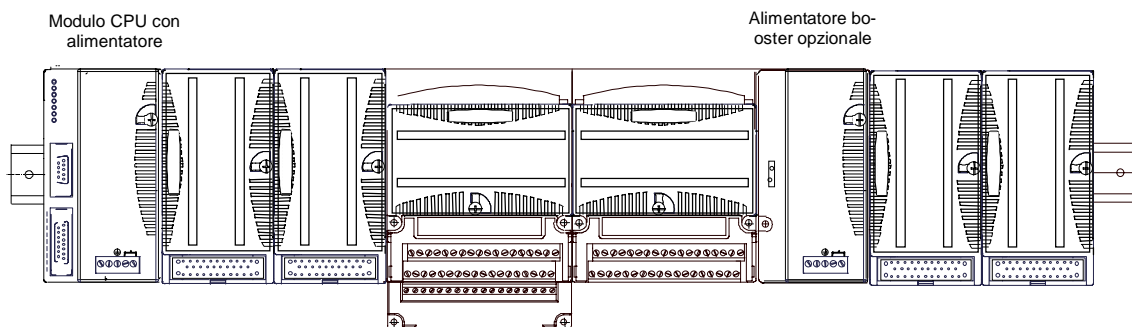
Manuale Utente Moduli VersaMax, Alimentatori e Basi (numero di catalogo GFK-1504)	Descrive i numerosi moduli di I/O e opzionali, gli alimentatori e le basi. Ciascun capitolo descrive i moduli di un particolare tipo. Il manuale fornisce anche istruzioni dettagliate per l'installazione.
Manuale Utente VersaMax Profibus NIU (numero di catalogo GFK-1534)	Descrive l'installazione e il funzionamento del Profibus NIU. Descrive anche il funzionamento dei moduli che possono essere presenti esclusivamente in un sistema NIU.
Manuale Utente VersaMax Genius NIU (numero di catalogo GFK-1535)	Descrive l'installazione e il funzionamento del Genius NIU. Descrive anche il funzionamento dei moduli che possono essere presenti esclusivamente in un sistema NIU.

Il PLC VersaMax™

La struttura compatta del PLC VersaMax™ è stata progettata per il montaggio su guide DIN. La CPU, grazie al suo potente set di istruzioni per la programmazione e alle sue caratteristiche avanzate, serve fino a otto moduli di I/O e opzionali, offrendo fino a 256 punti locali di I/O.

Nel sistema locale possono essere inclusi fino a otto moduli. L'alimentazione per il funzionamento dei moduli viene fornita da un alimentatore montato direttamente sulla CPU. Se necessario per moduli che richiedono elevata corrente, si potranno aggiungere ulteriori alimentatori.

I moduli di I/O vengono montati su supporti individuali, installati sulla guida DIN. Consentono comunicazioni sul backplane e il cablaggio eseguito in loco per i terminali del modulo. Vari supporti rendono flessibile montaggio e cablaggio.

La figura qui sotto mostra un sistema locale con sei moduli di I/O. I moduli possono essere montati su qualsiasi combinazione di supporti. Sui supporti del tipo a connettore, i moduli sono orientati verticalmente rispetto alla guida DIN. I supporti del tipo a connettore dispongono di un connettore a 32 pin per il collegamento di un cavo I/O. I terminali del cablaggio locale per questi supporti si trovano su speciali unità interposte (non nella figura). Sui supporti del tipo terminale, i moduli sono montati orizzontalmente rispetto alla guida DIN. I supporti del tipo terminale dispongono di 32 terminali a vite per la connessione diretta del cablaggio eseguito in loco. Se servono altri terminali, si potranno aggiungere terminali I/O ausiliari, uno dei quali è mostrato nella figura.

Progettato per l'automazione industriale e commerciale, VersaMax è conforme ai requisiti UL, CUL, CE, Classe 1 Zona 2 e Classe I Divisione 2.

L'autoconfigurazione all'accensione provvede all'indirizzamento automatico dei moduli del sistema. I moduli dispongono di funzioni preimpostate adatte a una vasta gamma di applicazioni.

Caratteristiche della CPU

La CPU VersaMax fornisce un vasto Set di Istruzioni, con programmazione nei formati Ladder Diagram e Sequential Function Chart. I programmi possono comprendere funzioni dati con virgola mobile (numeri reali).

La CPU con due porte seriali (IC200CPU001) offre eccellenti funzionalità PLC in un sistema piccolo e versatile. La CPU001 funziona come controllore di sistema per un massimo di 8 moduli con fino a 256 punti locali di I/O. La CPU dispone di due porte seriali incorporate per comunicazioni: RS232 e RS485. Entrambe le porte sono configurabili via software per operazioni slave SNP o RTU. Vengono supportati RTU a 4 e a 2 fili. Inoltre, la presenza della funzione di I/O seriale della CPU consente a un programma applicativo il controllo di attività lettura/scrittura di una porta per implementare la maggior parte dei protocolli seriali.

Memoria Programmi

I programmi applicativi vengono creati con il software di programmazione e trasferiti al PLC, quindi memorizzati nella memoria CPU con batteria tampone. La CPU001 dispone di 12KB di memoria per caricare i programmi applicativi.

Subroutine

Un programma può essere costituito da un programma principale che viene eseguito completamente durante lo sweep della CPU o da un programma suddiviso in subroutine. La dimensioni massima di un programma principale o di un blocco di subroutine per la CPU001 è di 12KB.

Le subroutine possono semplificare la programmazione e ridurre la quantità totale delle logiche. A ciascuna subroutine può essere assegnato un nome.

I blocchi delle subroutine possono essere bloccati e sbloccati dal software di programmazione, per variare i livelli di sicurezza del programma. Esistono quattro livelli di bloccaggio.

Riferimenti per le Condizioni del Sistema

La CPU VersaMax fornisce una gamma completa di riferimenti per le condizioni del sistema. La CPU aggiorna automaticamente queste locazioni di memoria, predefinite e con un nome, con informazioni sulle condizioni. Vi si può accedere, quando necessario, mediante le logiche del programma per controllare e reagire alle variazioni delle condizioni del sistema. Queste speciali referenze sulle condizioni ne comprendono del tipo a tempo definite T_10MS, T_100MS, T_SEC, e T_MIN e referenze di comodità del tipo FST_SCN (prima scansione), ALW_ON (sempre attiva), e ALW_OFF (sempre disattivata).

Set di Istruzioni

Il set di istruzioni della CPU VersaMax comprende l'intera gamma di funzioni comuni agli altri PLC GE Fanuc, come le Serie 90-30 e 90-70. Informazioni complete di riferimento sono disponibili nella relativa documentazione e nel Help in linea del software di programmazione.

Contatti Normalmente Aperto, Normalmente Chiuso	***Bobine*** Normalmente Aperto Negato Transizione Negative & Positiva SET / RESET Ritentivo SET / Ritentivo RESET Ritenzione Negata / Ritenzione
Timer e Contatori On-Delay Stopwatch Timer Off-Delay Timer On-Delay Timer Up Counter Down Counter	***Funzioni Matematiche*** Addizione / Sottrazione / Moltiplicazione / Divisione Modulo Divisione Trigonometria: Sene, Coseno, Tangente, Seno Inverso, Coseno Inverso, Tangente Inversa Conversione in Gradi, /Conversione in Radianti Radice Quadrata Logaritmi in Base 10 /Logaritmi naturali Spegnimento
Funzioni Relazionali Uguale Non Uguale Maggiore di Maggiore di o Uguale a Minore di Minore di o Uguale a Range	***Funzioni Bit*** AND / OR Logico / OR Esclusivo / Scorrimento a sinistra / Scorrimento a destra Rotazione a Sinistra / Rotazione a Destra Bit Test / Bit Set / Bit Annulla / Bit Posizione Comparazione mascherata
Funzione Dati Movimento Muovi Blocco Movimenti Annulla Blocco Scorrimento Registro Sequenziatore Bit Richiesta di Comunicazione	***Funzioni Tabella*** Array Move Ricerca: Uguale / Non Uguale / Maggiore di / maggiore di o Uguale a / Minore di / Minore di o Uguale a
Funzione Conversioni Conversione da intero a BCD4 Conversione da reale a word Convers. da BCD-4 o reale a intero Conversione BCD-4 o reale intero in doppia precisione Conversione intero, intero doppia peocisioneBCD-4 o reale in intero Tronca reale in intero Tronca reale in doppia precisione intero	***Funzioni di Controllo*** Chiamata Esegui I/O Algoritmo PID indipendente Algoritmo ISA PID Fine Logica Temporaneo Commenti Service Request Nested Master Control Relay Jump

Funzioni SVCREQ

La funzione Service Request, compresa nel set di istruzioni VersaMax, è un potente strumento di comunicazione che può essere utilizzato per consultare informazioni speciali sul funzionamento del sistema o per modificare certe operazioni con il programma applicativo.

Con le Service Request, il programma applicativo può:

- Modificare/leggere il Ciclo di Scansione Costante.
- Leggere la Finestra Tempi.
- Modificare il modo e il tempo delle finestre di programmazione e di comunicazione.
- Cambiare il modo e il tempo della finestra comunicazioni del sistema.
- Modificare/leggere le condizioni del compito Checksum e il numero delle Parole Checksum.
- Modificare/leggere l'Orologio dell'Ora del Giorno.
- Resettare il Timer Watchdog.
- Leggere il Tempo di Scansione dall'Inizio Scansione.
- Leggere il Nome del Folder.
- Leggere l'ID PLC.
- Leggere le Condizioni Operative PLC.
- Spegnere il PLC.
- Azzerare le Tabelle Errori.
- Leggere l'Ultimo Errore Immesso nella Tabella.
- Leggere il Clock Tempo Scaduto.
- Leggere le Condizioni Override I/O.
- Leggere la Checksum Master.
- Interrogare le I/O.
- Leggere la durata delle interruzioni di corrente.

I/O Seriale

Nel programma applicativo, può essere inserita un'altra funzione importante, la I/O seriale, per controllare le attività di lettura/scrittura di una delle porte CPU direttamente dal programma applicativo. Questa caratteristica può essere utilizzata per implementare la maggior parte dei protocolli seriali.

La I/O seriale può essere utilizzata per :

- Inizializzare una delle porte
- Impostare le dimensioni del buffer dati ricevuto
- Smistare il buffer dati ricevuto
- Verificare le condizioni della porta
- Attivare/disattivare l'RTS sulla porta seriale
- Cancellare un'operazione
- Far comporre a un modem un numero e inviare una specifica stringa di dati
- Inviare fino a 250 caratteri dalla memoria word a un dispositivo remoto mediante la porta specificata
- Leggere uno o più caratteri ricevuti da un buffer interno

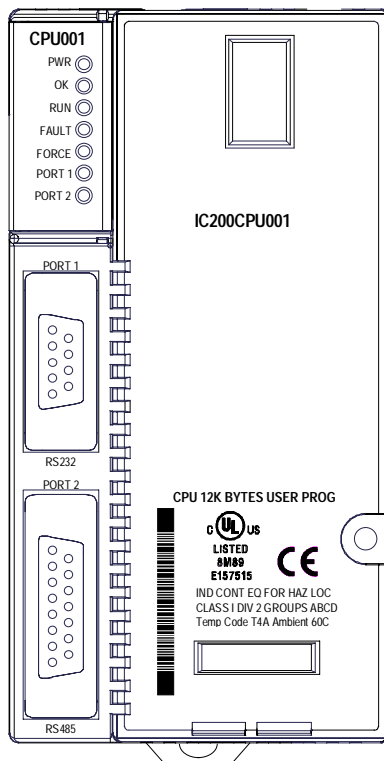
Capitolo 2

Moduli CPU

Questo capitolo descrive l'aspetto, le caratteristiche e il funzionamento dei moduli CPU per PLC VersaMax.

- IC200CPU001 CPU con due porte seriali

IC200CPU001
CPU con due porte seriali

La CPU con due porte seriali (IC200CPU001) consente un ottimo funzionamento del PLC, in un sistema piccolo e versatile. La CPU001 funge da controller del sistema per un massimo di 8 moduli con fino a 256 punti locali di I/O. La CPU ha due porte seriali: la porta 1 (RS-232) e la porta 2 (RS-485).

Caratteristiche

- RAM da 12 K con batteria tampone per i programmi applicativi
- Memoria flash non volatile per caricare i programmi
- Programmazione in Ladder Diagram, Sequential Function Chart, e Lista Istruzioni
- Batteria tampone per programmi, dati e ora del giorno
- Interruttore Avvio/Arresto
- Funzioni dati con virgola mobile (numeri reali)
- Comunicazioni incorporate mediante RS-232 e RS-485
- Altezza 70mm, quando montata su guida DIN con alimentatore

IC200CPU001
CPU con due Porte Seriali

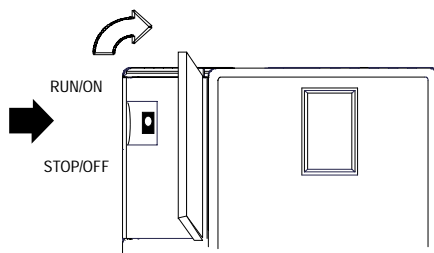
Specifiche del Modulo

Dimensioni	66.8mm x 128mm
Caricamento programma	Flash di sistema, RAM con batteria tampone
Virgola mobile	Si
Comunicazioni incorporate	RS-232, RS-485
Ambiente	*Vedere il manuale Moduli, Alimentatori e Supporti VersaMax* (GFK-1504).
Velocità di esecuzione boleana	1.8ms/K (tipico)
Accuratezza clock in tempo reale (temporizzazione funzioni)	100ppm (0.01%) o +/- 9sec/giorno
Accuratezza dell'ora del giorno	23ppm (0.0023%) o +/- 2sec/giorno @ 30C. 100 ppm (0.01%) o +/- 9sec/giorno @ nell'intera gamma di temperature

Commutatore Modo

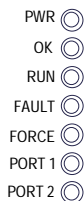
Il modulo CPU dispone di un comodo commutatore che può essere utilizzato per porre il PLC nel modo funzionamento o arresto (RUN/ON – STOP/OFF).

Lo stesso commutatore può essere utilizzato anche per bloccare scritture accidentali nella memoria CPU ed evitare forzature ai dati discreti. L'uso di questa funzione è configurabile.

La configurazione di base consente la selezione del modo funzionamento/arresto e disabilita la protezione della memoria.

IC200CPU001
CPU con due porte seriali

LED

Sette LED, visibili attraverso lo sportello, indicano la presenza di tensione, il modo operativo e le condizioni diagnostiche della CPU. Indicano inoltre la presenza di guasti, forzature e comunicazioni tramite le due porte del modulo.

PWR
OK
RUN
FAULT
FORCE
PORT 1
PORT 2

PWR Acceso quando la CPU è alimentata a 5V dall’alimentatore. Non indica le condizioni della tensione di alimentazione sul bus a 3,3V.

OK Acceso indica che la CPU ha superato la diagnostica iniziale e che funziona correttamente. Spento indica problemi a carico della CPU. Se lampeggia rapidamente, indica che la CPU sta eseguendo la diagnostica iniziale. Se lampeggia lentamente, indica che la CPU sta configurando moduli di I/O. Se questo LED lampeggia assieme al LED verde RUN significa che la CPU è nel modo boot in attesa di un aggiornamento firmware mediante la porta 1.

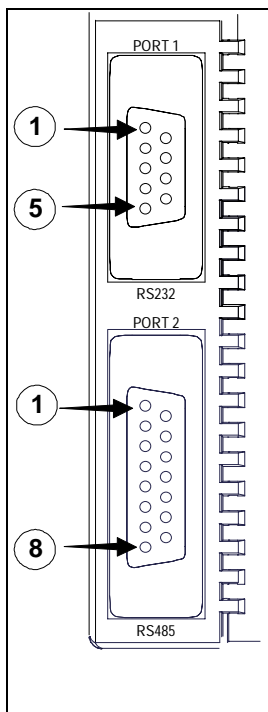
RUN Verde quando la CPU è nel modo funzionamento. Se giallo indica che la CPU è nel modo Stop/IO Scan. Se questo LED è spento, ma OK è acceso, la CPU è nel modo Stop/No IO Scan.

Se questo LED è verde e lampeggia e il LED FAULT è acceso, il commutatore del modulo è stato commutato da Stop a Run mentre si è manifestato un errore grave. Con una nuova commutazione continuerà in modo Run.

FAULT Acceso se la CPU è nel modo Stop a causa del manifestarsi di un grave errore. Per spegnere il LED FAULT, liberare sia la tabella guasti di I/O che la tabella guasti PLC. Se questo LED lampeggia e il LED OK è spento, è stato rilevato un errore grave durante la diagnostica iniziale del PLC. Chiamare l’assistenza tecnica.

FORCE Acceso se è attiva l’esclusione di un bit di riferimento.

PORT 1
PORT 2 Se lampeggia, indica attività su questa porta.

Porte Seriali CPU

Le due porte seriali della CPU possono essere configurate via software per operazioni slave SNP o RTU. Sono supportati RTU a 4 e a 2 fili. Quando una porta viene utilizzata per RTU, se necessario questa passa automaticamente al modo slave SNP. Entrambe le porte sono impostate sul modo slave SNP ed entrambe passano a questo modo quando la CPU è nel modo stop, se configurata per I/O Seriale.

Qualsiasi porta può essere configurata via software per impostare le comunicazioni tra la CPU e vari dispositivi seriali. Se richiede al massimo 100mA a 5VCC, un dispositivo interno può essere alimentato dalla porta 2.

Lunghezza Cavo e Baud Rate

Le lunghezze massime dei cavi sono la distanza tra la CPU e l'ultimo dispositivo applicato a un cavo.

Porta 1 (RS-232) = 15 metri
Porta 2 (RS-485) = 1200 metri

Entrambe le porte possono essere configurate in una gamma di Baud Rate da 4800 a 38,4K bps.

IC20CPU001
CPU con due porte seriali

Porta 1

La porta 1 è una porta RS-232 a 9 pin femmina con sub-connettore D. Viene utilizzata come porta di scaricamento per aggiornare il firmware della CPU. La schermatura del cavo rimane collegata alla conchiglia. I pin di uscita della porta 1 consentono la connessione diretta mediante un cavo a una porta RS-232 tipo AT.

Pin	Segnale	Direzione	Funzione
1	n/c		
2	TXD	Uscita	Trasmissione dati in Uscita
3	RXD	Entrata	Ricezione dati in Entrata
4	n/c		
5	GND	--	Segnale di riferimento 0V/terra
6	n/c		
7	CTS	Entrata	Pronto a ricevere in entrata
8	RTS	Uscita	Richiesta di Invio in uscita
9	n/c		
Conchiglia	SHLD	--	Connessione del cavo di schermatura (continuo) / 100% Connessione del cavo di schermatura

Porta 2

La Porta 2 è una porta RS-485 con 15-pin femmina e sub-connettore D. Può essere allacciata direttamente a un adattatore da RS-485 a RS-232 (IC690ACC901).

Pin	Segnale	Direzione	Funzione
1	SHLD	--	Connessione filo drenaggio cavo schermato
2, 3, 4	n/c		
5	P5V	Uscita	+5.1VCC aliment. dispositivi esterni(100mA max.)
6	RTSA	Uscita	Richiesta di invio (A) Uscita
7	GND	--	Segnale di riferimento a terra
8	CTSB’	Entrata	Pronto a ricevere (B) Entrata
9	RT	--	Terminazione resistenza (120 ohm) per RDA’
10	RDA’	Entrata	Ricezione dati (A) in ingresso
11	RDB’	Entrata	Ricezione dati (B) in ingresso
12	SDA	Uscita	Trasmissione dati (A) Uscita
13	SDB	Uscita	Trasmissione dati (B) Uscita
14	RTSB	Uscita	Richiesta di invio (B) in Uscita
15	CTSA’	Entrata	Pronto a ricevere (A) in entrata
Involucro	SHLD	--	Connessione del cavo di schermatura (continuo) / 100% Connessione del cavo di schermatura

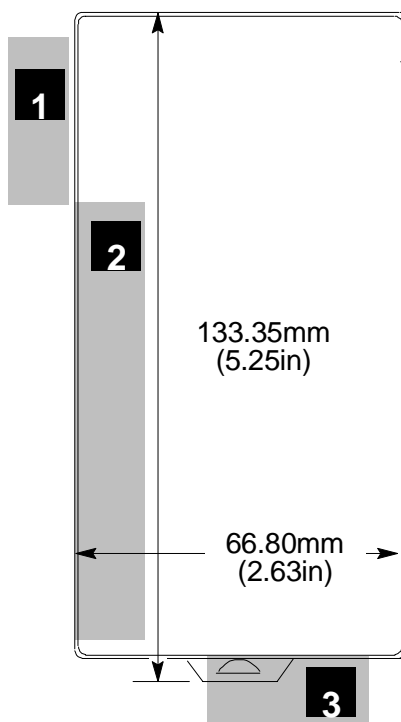
Capitolo 3 Installazione

Questo capitolo spiega come installare la CPU.

- Installazione della CPU
- Installazione dell'alimentatore
- Installazione di moduli aggiuntivi
- Attivazione o sostituzione della batteria tampone
- Connessioni della porta seriale per la CPU001
- Requisiti per l'installazione secondo la normativa CE

Le istruzioni per l'installazione del sistema spiegano come installare il supporto, l'alimentatore e i moduli, mentre le informazioni riguardanti il cablaggio locale e la messa a terra si trovano sul *Manuale per i moduli, l'alimentatore e le basi di I/O VersaMax*, GFK-1504.

Installazione della CPU

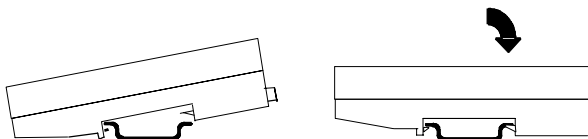
Lasciare uno spazio di 5cm sopra e sotto e uno spazio di 2,5cm sulla sinistra. Ulteriori spaziature verranno indicate in seguito. I moduli vanno montati su di una guida DIN orizzontale.

1	Lasciare sufficiente spazio per l’apertura dello sportello della CPU.
2	Lasciare sufficiente spazio per i cavi della porta Seriale.
3	Lasciare sufficiente spazio per il cablaggio di alimentazione.

Installazione della CPU sulla guida DIN

La CPU e i supporti devono essere installati su di un'unica sezione di una guida DIN di 7,5mm x 35mm, che deve essere messa a terra per fornire la protezione EMC. La guida deve avere una finitura conduttiva (non verniciata) e resistente alla corrosione. Sono preferibili le guide DIN conformi a DIN EN50032.

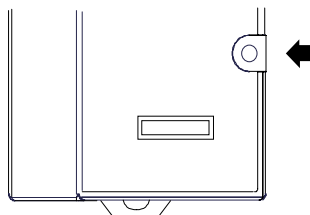
Per evitare vibrazioni, la guida DIN dovrebbe essere installata su di un pannello con viti spaziate di circa 5,24cm. Si possono anche installare dei fermi per la guida DIN (numero di catalogo IC200ACC313) su entrambi i lati dell'unità per bloccare in posizione i moduli.

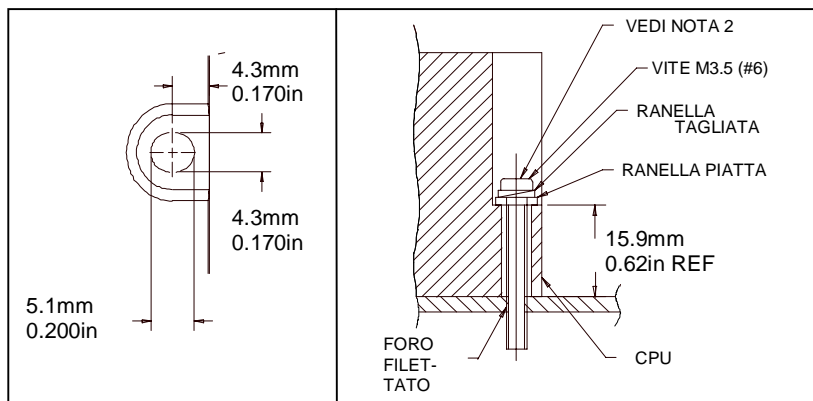
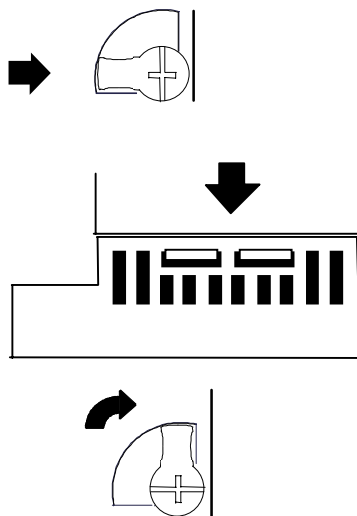
Per applicazioni che richiedono la massima resistenza alle vibrazioni e agli urti, anche la CPU e i supporti montati sulla guida DIN dovrebbero essere installati sul pannello. I fori di montaggio del pannello possono essere localizzati utilizzando il supporto come dima, o in base alle dimensioni indicate nel *Manuale Moduli, Alimentatori e Supporti VersaMax* (GFK-1504). Predisporre i fori di montaggio, quindi installare la CPU e i supporti con viti M3.5 (#6).

La base si aggancia facilmente alla guida DIN e non servono attrezzi per il montaggio sulla guida e per la messa a terra.

Rimozione della CPU dalla guida DIN

1. Togliere corrente all'alimentatore.
2. Se la CPU è applicata al pannello con delle viti, rimuovere il modulo di alimentazione, quindi rimuovere le viti di montaggio dal pannello.
3. Far scorrere la CPU lungo la guida DIN allontanandola dagli altri moduli fino a che si sblocca il connettore.
4. Con un piccolo cacciavite a testa piatta, tirare verso l'esterno l'aggancio alla guida DIN, inclinando l'altro lato del modulo verso il basso per sganciarlo dalla guida DIN.

Montaggio a pannello

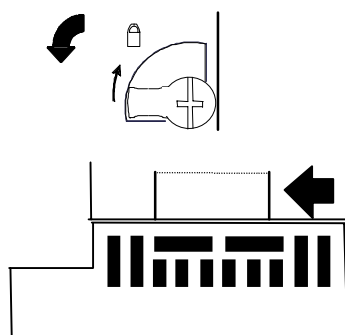
Per la massima resistenza meccanica a vibrazioni e urti, anche il modulo montato su guida DIN deve essere installato su di un pannello. Usando il modulo come modello, segnare sul pannello la posizione dei fori per montare il modulo sul pannello stesso, quindi forare il pannello. Installare il modulo mediante i fori praticati sul pannello con viti M3.5 (#6).

Nota 1. La tolleranza (non cumulabile) su tutte le dimensioni è di +/- 0.13mm.

Nota 2. Applicare una coppia da 1.1-1.4Nm (10-12 in/lbs) a viti in acciaio M3.5 (#6-32) in materiale filettato con spessore minimo di 2,4 mm.

Installazione dell'alimentatore

1. Il modulo di alimentazione viene montato direttamente sopra la CPU. La chiavetta dell'alimentatore deve rimanere sbloccata.

2. Allineare i connettori e il piedino della serratura, quindi premere sull'alimentatore fino a che i due innesti scatteranno in posizione. Verificare che questi siano ben inseriti nelle loro sedi predisposte sul lato inferiore della CPU.

3. Ruotare la chiavetta in posizione di chiusura per bloccare l'alimentatore sopra il modulo della CPU.

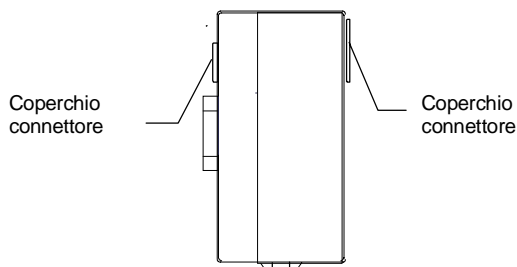
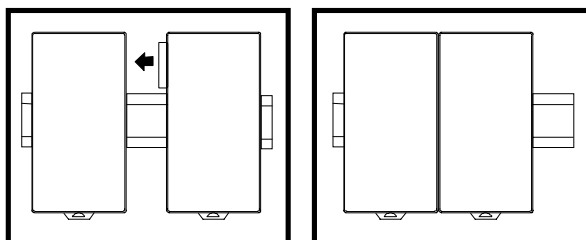
Rimozione dell'Alimentatore

Prestare attenzione quando si lavora vicino a parti in funzione. Alcuni dispositivi potrebbero essere molto caldi e provocare ustioni.

1. Togliere corrente.
2. Sbloccare la chiavetta, come indicato nella figura a lato.
3. Premere la linguetta flessibile sul lato inferiore dell'alimentatore per liberare gli innesti dalle sedi predisposte sulla base.
4. Tirare l'alimentatore per staccarlo.

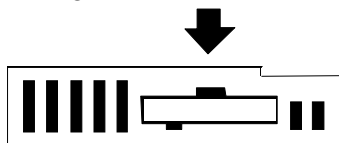
Installazione di moduli aggiuntivi

La CPU può gestire fino a 8 moduli aggiuntivi di I/O e opzionali.

Prima di aggiungere supporti alla CPU, rimuovere il coperchio del connettore sul lato destro della CPU. Conservare il coperchio, che dovrà essere installato sull'ultimo supporto, per proteggere i pin del connettore da danneggiamenti e ESD durante la manipolazione e l'uso.

Non rimuovere il coperchio del connettore del lato sinistro.

Installare i moduli aggiuntivi montandoli sui relativi supporti e facendoli scorrere lungo la guida DIN fino ad agganciare completamente i connettori sui lati dei supporti.

La CPU alimenta a +5V e +3,3V i moduli a valle tramite i connettori di accoppiamento. La quantità di moduli che possono essere supportati dipende dai requisiti di corrente dei vari moduli. Per fornire la corrente necessaria a tutti i moduli si potranno utilizzare alimentatori ausiliari. Il software di configurazione esegue i calcoli per la corrente richiesta da una configurazione hardware valida.

Attivazione o Sostituzione di una Batteria Tampone

Il modulo CPU viene fornito con una batteria già installata. L'alloggiamento della batteria si trova sul lato superiore del modulo CPU. Prima dell'uso iniziale, attivare la batteria sfilando la linguetta di isolamento.

Sostituzione dalla Batteria al Litio

Per sostituire la batteria, aprire l'alloggiamento della batteria con un piccolo cacciavite.

Per la sostituzione, usare soltanto batterie dei seguenti tipi:

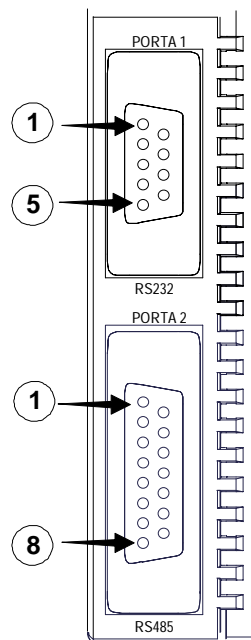
GE Fanuc	IC200ACC001
Panasonic	BR2032

Batterie di altro tipo potrebbero creare rischi di incendio o esplosione.

Attenzione

Se non trattata correttamente, la batteria può esplodere.

Non va ricaricata, aperta, riscaldata oltre 100°C o incenerita.

Connessioni della porta seriale per la CPU001

Alimentazione di un Dispositivo Esterno dalla Porta 2

Se una delle porte è impostata per comunicare con un dispositivo seriale che non richiede più di 100mA a 5VCC, questo può essere alimentato dalla Porta 2.

Lunghezze dei Cavi e Baud Rate

Le lunghezze massime dei cavi (lunghezza massima dalla CPU all'ultimo dispositivo collegato ai cavi) sono:

Porta 1 (RS-232) = 15 metri
Porta 2 (RS-485) = 1200 metri

Entrambe le porte possono essere configurate per un range di baud rate compreso tra 4800 e 38,4k bps.

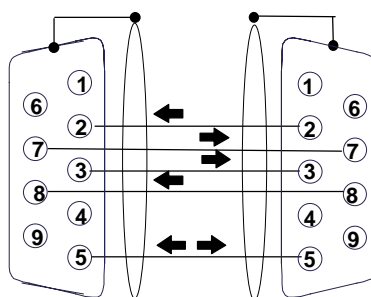
CPU001 Porta 1: RS-232

Assegnazione dei Pin per la Porta 1

La Porta 1 è una porta RS-232 con un subconnettore femmina a 9-pin tipo D. Viene utilizzata come porta di scaricamento per aggiornare il firmware della CPU. I pin di uscita della Porta 1 permettono di collegare direttamente un cavo con una porta standard RS-232 tipo AT. La schermatura del cavo è collegata alla conchiglia.

Pin	Segnale	Direzione	Funzione
1	n/c		
2	TXD	Uscita	Trasmissione dati in uscita
3	RXD	Entrata	Ricezione dati in entrata
4	n/c		
5	GND	--	Segnale di riferimento 0V/TERRA
6	n/c		
7	CTS	Entrata	Pronto a ricevere in entrata
8	RTS	Uscita	Richiesta di invio in uscita
9	n/c		
Involucro	SHLD	--	Connessione cavo di schermatura / 100% (Continuo) connessione del cavo di schermatura alla schermatura

RS-232 Connessione Punto a Punto

Nella configurazione punto a punto, due dispositivi sono connessi sulla stessa linea di comunicazione. Per la RS-232, la lunghezza massima è di 15 metri.

PC 9-Pin Porta Seriale	CPU Porta 1
9-pin femmina	9-pin maschio
(2) RXD	(2) TXD
(3) TXD	(3) RXD
(5) GND	(5) GND
(7) RTS	(7) CTS
(8) CTS	(8) RTS

La schermatura deve essere connessa alla conchiglia dei connettori su entrambi i lati del cavo.

CPU001: Specifiche del Connettore e del Cavo per la Porta 1

I numeri di catalogo riportati qui sotto vanno intesi come riferimento. Si può utilizzare qualsiasi componente che risponda alle stesse specifiche.

<table>
<tr><td>Cavo:
Belden
9610</td><td colspan="4">Cavo computer, completamente intrecciato su lamina di schermatura
5 conduttori †
30 Volt / 80°C
24 AWG rame stagnato, 7x32 trefoli</td></tr>
<tr><td rowspan="2">Connettore maschio a 9 pin:</td><td><u>Tipo:</u>
Crimp</td><td><u>Fornitore:</u>
ITT/Cannon
AMP</td><td><u>Spinotto:</u>
DEA9PK87F0
205204-1</td><td><u>Pin:</u>
030-2487-017
66506-9</td></tr>
<tr><td>Solder</td><td>ITT/Cannon
AMP</td><td>ZDE9P
747904-2</td><td>--
--</td></tr>
<tr><td>Connettore
Involucro:</td><td colspan="4">Kit *– ITT Cannon DE121073-54 [formato 9-pin backshell kit]:
Plastica metallizzata (Plastica con nichel su rame) †
Attacco cavo di terra (compreso)
Uscita cavo a 40° per mantenere compatta l’installazione
Più – ITT Cannon 250-8501-010 [Jackscrew sporgente]:
Filetto #4-40 per un attacco scuro alla porta della CPU †
Ordinarne 2 per ogni cavo involucro</td></tr>
</table>

† Informazione critica – qualsiasi altra parte scelta deve rispondere a questi criteri.

* L’uso di questo kit mantiene a 70 mm la profondità di installazione.

CPU001 Porta 2: RS-485

Assegnazione Pin per la Porta 2

La Porta 2 è una porta RS-485 con subconnettore D femmina a 15-pin. Può essere collegata direttamente a un adattatore da RS-485 a RS-232.

Pin	Segnale	Direzione	Funzione
1	SHLD	--	Connessione drenaggio cavo schermato
2, 3, 4	n/c		
5	P5V	Uscita	+5.1VCC per alimentare dispositivi esterni (100mA max.)
6	RTSA	Uscita	Richiesta di invio (A) in uscita
7	GND	--	Segnale di riferimento 0V/TERRA
8	CTSB'	Entrata	Pronto a ricevere (B) in entrata
9	RT	--	Terminazione resistenza (120 ohm) per RDA'
10	RDA'	Entrata	Ricezione dati (A) in entrata
11	RDB'	Entrata	Ricezione dati (B) in entrata
12	SDA	Uscita	Trasmissione dati (A) in uscita
13	SDB	Uscita	Trasmissione dati (B) in uscita
14	RTSB	Uscita	Richiesta di invio (B) in uscita
15	CTSA'	Entrata	Pronto a ricevere (A) in entrata
Involucro	SHLD	--	Connessione cavo di schermatura / 100% (Continuo) conness. cavo di schermatura alla schermat.

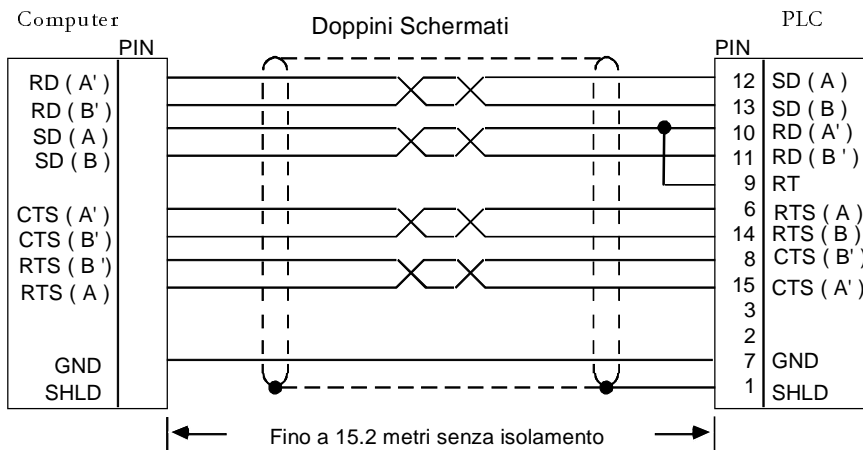
Specifiche del Connettore e del Cavo per la Porta 2

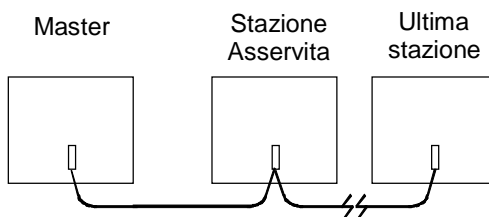
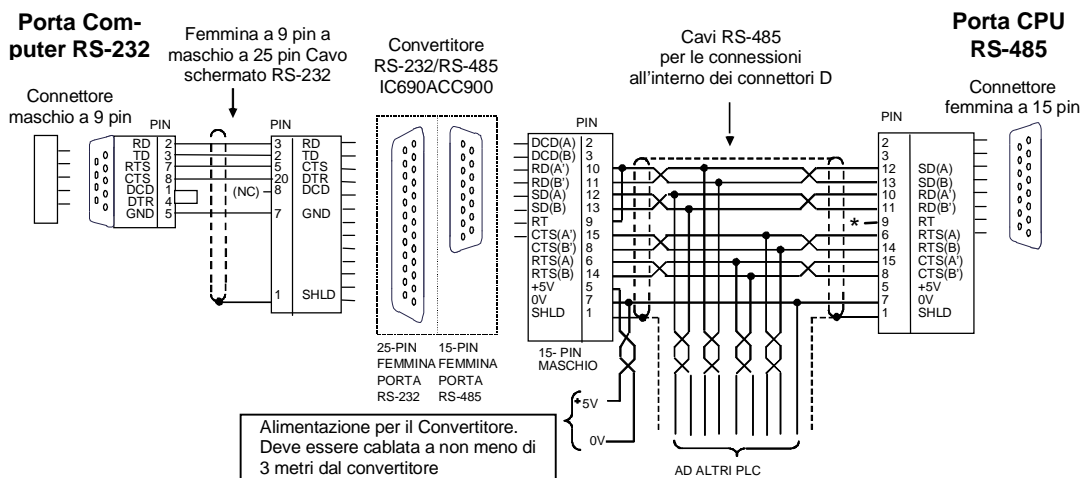
I numeri di catalogo riportati qui sotto vanno intesi come riferimento. Si può utilizzare qualsiasi parte che risponda alle stesse specifiche.

Cavo: Belden 8105	Cavo computer di basa Capacitanza, completamente intrecciato su lamina di schermatura 5 Doppini † Cavo drenaggio Involucro † 30 Volt / 80°C (176°F) 24 AWG rame zincato, 7x32 trefoli Velocità di propagazione = 78% Impedenza nominale = 100Ω †			
Connettore maschio a 15 pin:	<u>Tipo:</u> Crimp	<u>Fornitore:</u> ITT/Cannon AMP	<u>Spinotto:</u> DAA15PK87F0 205206-1	<u>Pin:</u> 030-2487-017 66506-9
	Solder	ITT/Cannon AMP	ZDA15P 747908-2	-- --
Connettore Involucro:	Kit* – ITT Cannon DA121073-50 [formato 15-pin backshell kit]: Plastica metallizzata (Plastica con nichel su rame) † Attacco cavo di terra (compreso) Uscita cavo a 40° per mantenere compatta l'installazione Più – ITT Cannon 250-8501-009 [Jackscrew sporgente]: Filetto metrico #M3x0,5 per un attacco scuro † Ordinarne 2 per ogni cavo involucro			

† Informazione critica – qualsiasi altra parte scelta deve rispondere a questi criteri.

CPU001: RS485 Connessione Punto a Punto con Handshaking

Nella connessione punto a punto, due dispositivi sono collegati alla stessa linea di comunicazione. Per la RS-485, la lunghezza massima del cavo è di 1200 metri. Per distanze superiori si possono utilizzare dei modem.

CPU001: connessioni seriale RS-485 multidrop

Nella configurazione a cascata, il dispositivo host viene configurato come master e uno o più PLC vengono configurati come slave. La distanza massima tra il master e i dispositivi slave non può superare i 1200 metri. Si presume che i cavi siano di buona qualità e il rumore presente sia moderato. Con configurazione daisy chain o multidrop, alla RS-485 si possono connettere fino a 8 asservimenti. La linea RS-485 deve includere un handshaking e devono essere usati i conduttori prima specificati.

Nel cablaggio RS-485 a cascata, le riflessioni di segnale sulla linea di trasmissione possono essere ridotti dal cablaggio a *ghirlanda* indicato qui sotto. Eseguire le connessioni all'interno del connettore da applicare al PLC. Evitare l'uso di terminazioni con altri tipi di connettori lungo la linea di trasmissione.

La resistenza di terminazione per il segnale ricezione dati (RD) deve essere connessa solo alle unità ai capi delle linee. Questa terminazione viene eseguita sulla CPU con un jumper tra il pin 9 e il pin 10 all'interno del connettore D.

Potenziale di terra: perché il sistema funzioni correttamente, unità multiple non collegate alla medesima alimentazione devono avere lo stesso potenziale comune di terra o l'isolamento da terra.

Requisiti per Installazioni con Marchio CE

Le applicazioni che richiedono il marchio CE, devono rispondere ai seguenti requisiti di protezione contro scariche elettrostatiche (ESD) e scariche transienti veloci (FBT).

- Il PLC VersaMax è considerato un dispositivo aperto, pertanto va installato in un alloggiamento (IP54).
- Si prevede che questa apparecchiatura venga installata in ambienti tipicamente industriali che utilizzano materiali antistatici, come pavimenti in cemento o legno. Se utilizzata in ambienti che contengono materiali che possono creare cariche statiche, come i tappeti, il personale dovrà scaricarsi impugnando saldamente una struttura messa a terra prima di accedere a questo dispositivo.
- Se gli I/O sono alimentati dalla rete CA, prima della distribuzione agli I/O tali linee dovrebbero essere protette da soppressore, in modo da non eccedere i livelli di immunità dell'I/O. La soppressione per l'alimentazione CA degli I/O si può ottenere con dei MOV line-rated connessi tra linea e linea o tra linea e terra. Si deve provvedere a una buona connessione a terra ad alta frequenza per i MOV linea-terra.
- Si intende che le alimentazioni CA o CC inferiori a 50V vengano fornite localmente dalla rete CA. La lunghezza massima del cablaggio tra tali punti di alimentazione e il PLC non deve superare i 10 metri.
- L'installazione deve essere all'interno con l'alimentazione CA primaria protetta dalle scariche.
- In presenza di disturbi, la comunicazione seriale potrebbe interrompersi.

Capitolo 4

Configurazione

Questo capitolo descrive il processo mediante il quale vengono configurati la CPU VersaMax e i moduli a essa asserviti. La configurazione determina certe caratteristiche di funzionamento del modulo e stabilisce anche i riferimenti al programma che verranno utilizzati da ciascun modulo del sistema.

La CPU supporta sia l'autoconfigurazione che il caricamento di una configurazione da un sistema di programmazione.

Caricamento di una configurazione da un sistema di programmazione

Una configurazione può essere caricata da un programmatore tramite la porta CPU.

Il caricamento di una configurazione disabilita l'autoconfigurazione.

Nota: Se sulla CPU è caricata una configurazione hardware, la configurazione per la porta seriale al quale è collegato il programmatore non sarà effettivamente installata fino a quando verrà scollegato il programmatore. Dopo lo scollegamento del programmatore, il nuovo protocollo diverrà operativo con un certo ritardo; un ritardo pari al tempo configurato in T3'.

L'eliminazione di una configurazione eseguita da un programmatore causerà la creazione di una nuova autoconfigurazione, che rimarrà abilitata fino a che verrà caricata di nuovo una configurazione dal programmatore.

Autoconfigurazione

Quando è abilitata un'autoconfigurazione e non ne esiste una precedente, all'accensione la CPU leggerà automaticamente la configurazione dei moduli installati sul sistema e creerà una configurazione generale del sistema stesso. Se all'accensione è presente un'autoconfigurazione precedente, la configurazione verrà elaborata come descritto in Diagnostica dell'Autoconfigurazione.

I moduli con caratteristiche configurabili via software, quando autoconfigurati assumono le loro impostazioni di default. Queste caratteristiche sono descritte nel *Manuale Moduli, Alimentatori, e Supporti VersaMax* (GFK-1504).

"Slot"

Ciascun I/O o modulo opzionale occupa uno "slot" del sistema. Il modulo più vicino alla CPU occupa lo slot 1. Gli alimentatori aggiuntivi non occupano slot.

All'accensione, la CPU per default automaticamente genera una configurazione che comprende tutti i moduli fisicamente presenti sul sistema, partendo dallo slot 1. La configurazione si ferma al primo slot vuoto o a un modulo guasto. Per esempio, se vi sono moduli fisicamente presenti soltanto negli slot 1, 2, 3, 5 e 6, i moduli negli slot 5 e 6 non vengono configurati.

Il modo in cui l'autoconfigurazione tratta i moduli aggiunti, rimossi o guasti viene descritto nella prossima pagina.

L'Autoconfigurazione Assegna gli Indirizzi di Riferimento

Ai moduli viene automaticamente assegnato un indirizzo di riferimento in ordine crescente. Per esempio, se il sistema contiene due moduli input da 16 e da 8 punti, un modulo output da 16 punti e un altro modulo input da 16 punti, in questo ordine, ai moduli input vengono assegnati indirizzi di riferimento di %I0001, %I0017, e %I0025, rispettivamente. Per i moduli che utilizzano tipi di dati multipli (es. moduli di I/O misti), a ciascun tipo di dato viene assegnato un indirizzo di riferimento individuale.

Diagnostica dell'Autoconfigurazione

Modulo presente ma non operativo durante l'autoconfigurazione: Se un modulo è fisicamente presente, ma non operativo durante l'autoconfigurazione, questo modulo non viene configurato e la CPU genera una diagnostica *modulo extra*.

Slot vuoto durante l'autoconfigurazione: L'autoconfigurazione si ferma al primo slot vuoto, e i moduli successivi non vengono configurati. Per ciascuno di questi la CPU genera una diagnostica *modulo extra*.

Moduli precedentemente configurati presenti durante l'autoconfigurazione: I moduli precedentemente configurati non vengono rimossi dalla configurazione durante l'autoconfigurazione, a meno che non siano più presenti sul sistema. Per esempio, se vengono configurati moduli sugli slot 1, 2 e 3, poi viene tolta corrente e viene rimosso il modulo 1, alla riaccensione i moduli 2 e 3 verranno configurati normalmente. Il modulo originariamente nello slot 1 non viene rimosso dalla configurazione e la CPU genera una diagnostica *modulo mancante* per lo slot 1.

Moduli diversi presenti durante l'autoconfigurazione: Se uno slot era stato precedentemente configurato per un tipo di modulo, ma durante l'autoconfigurazione ne viene scoperto uno diverso, la CPU genera una diagnostica *errore di mismatch nella configurazione*. Lo slot rimane configurato per il tipo di modulo originale.

Modulo non configurato installato dopo l'autoconfigurazione: Se un modulo non precedentemente configurato viene aggiunto dopo l'accensione, la CPU genera una diagnostica *modulo extra* e il modulo non viene aggiunto alla configurazione.

Modulo precedentemente configurato installato dopo l'autoconfigurazione: Se all'accensione manca un modulo precedentemente configurato e questo viene aggiunto dopo l'accensione, la CPU genera una diagnostica *aggiunta di un modulo* e il modulo viene aggiunto in coda nella scansione I/O.

Rimozione di tutti i moduli dopo l'autoconfigurazione: Se all'accensione mancano tutti i moduli, la CPU cancella la configurazione. Ciò consente di inserire dei moduli che verranno configurati alla successiva accensione.

Sommario Messaggi Diagnostici dell'Autoconfigurazione

addition of module (aggiunta di un modulo)	1. un modulo è presente all'accensione, ma non configurato. 2. l'autoconfigurazione è abilitata e il modulo può essere configurato.
addition of module (aggiunta di un modulo)	un modulo precedentemente mancante viene inserito dopo l'accensione. Se il modulo era stato configurato, la CPU riprende la scansione del modulo. Diversamente, il modulo non viene aggiunto alla configurazione e non viene scandito.
configuration mismatch (configurazione non conforme)	all'accensione o dopo l'accensione viene scoperto un modulo non conforme alla configurazione dello slot.
extra module (modulo extra)	1. all'accensione è presente un modulo non configurato. 2. l'autoconfigurazione non è abilitata.
loss of module (modulo mancante)	durante l'accensione o il normale funzionamento manca un modulo configurato.
unsupported feature (caratteristica non supportata)	è presente un modulo non supportato dalla CPU.

Capitolo 5 Funzionamento della CPU

Questo capitolo descrive i modi operativi della CPU per PLC e indica la relazione tra l'esecuzione del programma applicativo e gli altri compiti eseguiti dalla CPU.

Modi Oprativi della CPU

In un PLC, il programma applicativo viene eseguito ripetutamente. Oltre a eseguire il programma applicativo, la CPU per PLC riceve regolarmente dati dai dispositivi input, invia dati ai dispostivi output, esegue la gestione interna e assolve i compiti di comunicazione. Questa sequenza operativa viene chiamata **sweep**.

- Il modo operativo standard del PLC viene chiamato modo **Sweep Standard**. In questo modo, la CPU esegue normalmente ogni componente dello sweep. Ciascun sweep viene eseguito il più rapidamente possibile, anche se i vari sweep richiedono tempi di esecuzione diversi.
- La PLC può però operare anche nel modo a **Tempo di Sweep Costante**. In questo modo, la CPU esegue la stessa serie di azioni, ma ciascun sweep avviene nello stesso intervallo di tempo
- La PLC può anche essere in uno sei seguenti modi Stop:
  - Stop con il modo I/O Disabilitato
  - Stop con il modo I/O Abilitato

Schema Sweep della CPU

Schema Sweep della CPU

Inizio della Gestione Sweep	La gestione comprende i compiti necessari per preparare l'avvio dello sweep. Prima di iniziare lo sweep effettivo, la CPU: Calcola il tempo Sweep Programma l'avvio dello sweep successivo Determina il modo dello sweep successivo Aggiorna la tabella di riferimento degli errori Azzera il timer Watchdog Se il PLC è nel modo Tempo di Sweep Costante, lo sweep viene ritardato fino allo scadere del tempo sweep richiesto. Se tale tempo è già scaduto, viene impostato il contatto OV_SWP %SA0002 e lo sweep continua senza ritardi. Poi, la CPU aggiorna i valori del timer (centesimi, decimi e secondi).
Scansione Dati Ricevuti	Quando inizia lo sweep, la CPU prima scansiona i dati ricevuti dai moduli input e da quelli opzionali. I moduli vengono scansionati in ordine crescente rispetto all'indirizzo di riferimento. I moduli input discreti vengono scansionati prima dei moduli input analogici. La CPU carica i nuovi dati ricevuti nelle appropriate memorie discrete. Se la CPU è stata configurata per non scansionare nel modo I/O Stop, quando la CPU e nel nodo stop, la scansione input viene saltata.
Scansione Logica Programma Applicativo	Quindi, la CPU svolge la logica del programma applicativo, iniziando sempre con la prima istruzione del programma e terminando quando viene eseguito il comando END (fine). Lo svolgimento della logica crea un nuovo set di dati output.
Scansione Dati Inviati	Immediatamente dopo lo svolgimento della logica, la CPU scansiona tutti i moduli output in ordine crescente rispetto all'indirizzo di riferimento. La scansione dei dati in uscita finisce quando termina l'invio di tutti i dati in uscita. Se la configurazione della CPU è non scansione I/O nel modo Stop, quando la CPU e nel nodo stop, anche la scansione output viene saltata.
Finestra di Comunicazione col Programma	Se è installato un dispositivo di programmazione, successivamente la CPU esegue la finestra di comunicazione con il programmatore. Questa non viene invece eseguita se non è installato né un programmatore né alcuna scheda da configurare nel sistema. Con ogni sweep viene configurata una sola scheda. Nel modo di default finestra limitata, a ciascun sweep la CPU rispetta una service request. Il tempo limite di comunicazione con il programmatore è di 6 msec. Se il programmatore esegue una richiesta la cui elaborazione richiede più di 6 msec., la richiesta viene suddivisa in vari sweep.
Finestra di Comunicazione col Sistema	Quindi, la CPU elabora le richieste di comunicazione dei moduli opzionali intelligenti. I moduli vengono assortiti a richiesta, in modo che nessun modula sia prioritario. Nel modo di default ("Run to Completion"), la durata della finestra di comunicazione col sistema è limitata a 400 msec. Se un modulo esegue una richiesta la cui elaborazione richiede più di 400 msec., la richiesta viene suddivisa in vari sweep. Nel modo limitato, i moduli opzionali che comunicano con la CPU mediante la finestra di sistema influiscono meno sul tempo di sweep, ma la risposta alla loro richiesta è più lenta.
Calcolo Checksum Programma Logico	Al termine di ciascun sweep. viene eseguito il calcolo Checksum sul programma applicativo. Per la Checksum si possono specificare da 0 a 32 word. Se la Checksum calcolata non corrisponde alla Checksum di riferimento, viene attivato il flag di errore di Checksum del programma, che inserirà un errore nella

	tabella errori PLC; inoltre, il modo PLC viene commutato su Stop. Se il calcolo Checksum fallisce, la finestra di comunicazione col programmatore non viene influenzata.

Sweep Standard della CPU

Lo Sweep Standard è il normale nodo operativo della CPU per PLC, con il quale la CPU esegue ripetutamente il programma applicativo, aggiorna gli I/O, oltre a eseguire le comunicazioni e gli altri compiti indicati nello schema:

1. La CPU esegue i compiti di gestione di avvio dello sweep.
2. Legge i dati in arrivo.
3. Esegue il programma applicativo.
4. Aggiorna i dati in uscita.
5. Comunica con un dispositivo di programmazione, se presente.
6. Comunica con altri dispositivi.
7. Esegue la diagnostica

Esclusa la comunicazione con il programmatore, tutti questi punti vengono eseguiti a ogni sweep. La comunicazione col programmatore avviene solo quando necessaria.

In questo modo, la CPU esegue normalmente tutte le sue fasi sweep. Ciascun sweep viene eseguito il più rapidamente possibile, anche se con tempi diversi per ciascun sweep.

Le Finestre Sweep

La finestra di comunicazione col programmatore e la finestra di comunicazione col sistema dispongono di due modi operativi:

Modo Limitato	Il tempo di esecuzione della finestra è 6 msec. Le finestra termina all'esaurimento compiti da assolvere o allo scadere dei 6 msec.
Modo Esecuzione Completa	Indipendentemente dal tempo assegnato a una certa finestra, la finestra opera fino al termine dei compiti in essa contenuti (fino a 400ms).

SVCREQ 2 può essere utilizzato nel programma applicativo per ottenere il tempo effettivo di ciascuna finestra.

Il Timer Watchdog

Quando la CPU è nel modo **Sweep Standard**, il Timer Watchdog rileva le condizioni di errore che potrebbero causare sweep insolitamente lunghi. Il timer Watchdog è impostato a 500 msec., che vengono contati a partire da zero all'inizio di ciascun sweep.

Se lo sweep dura oltre 500 msec., il LED OK sul modulo della CPU si spegne. La CPU si resetta, esegue la logica di accensione, genera un errore Watchdog e passa al modo Stop. Le comunicazioni vengono temporaneamente interrotte.

Sweep a Tempo Costante

Se l'applicazione richiede che ciascun sweep della CPU avvenga nello stesso tempo, la CPU può essere configurata per operate a Tempo di Sweep Costante. Questo modo operativo garantisce che nel sistema i dati in ingresso e in uscita vengano aggiornati a intervalli costanti. Questo modo può essere utilizzato anche per implementare un tempo sweep più lungo, per garantire che i dati in ingresso abbiano il tempo di disporsi dopo che saranno stati ricevuti i dati in uscita dal programma.

Modifica del Default Configurato per il Modo Sweep Costante

Se il PLC è nel modo STOP, il suo Modo Sweep Costante configurato può essere modificato. Eseguita questa modifica, perché abbia effetto la configurazione deve essere memorizzata nella CPU. Dopo la memorizzazione, il modo Sweep a Tempo Costante diventa il modo sweep di default.

Il Timer pere lo Sweep Costante

Durante il funzionamento nel modo Sweep a Tempo Costante, il Timer per il Tempo Costante della CPU controlla la durata dello sweep. Il timer può essere impostato da 5 a 500 msec. Per evitare errori di oversweep, il tempo impostato deve essere superiore di almeno 10 msec. al tempo sweep della CPU quando questa è nel modo Sweep Standard.

Se il timer per lo Sweep Costante interviene prima che termini lo sweep, la CPU terminerà comunque l'intero sweep, finestre comprese, segnalando però automaticamente che è avvenuto uno sweep troppo lungo. Nello sweep successivo a quello troppo lungo, la CPU introdurrà un allarme di oversweep nella tabella errori del PLC. Poi, all'inizio dello sweep successivo, la CPU imposta un errore di contatto OV_SWP (%SA0002). La CPU resetterà automaticamente il contatto OV_SWP quando il tempo di sweep non supererà più quello del Timer per lo Sweep Costante. La CPU resetterà il contatto OV_SWP anche se non opera nel modo Sweep a Tempo Costante.

Come per gli altri errori di contatto, il programma applicativo può controllare questo contatto per segnalare che si è manifestato uno sweep troppo lungo.

Abilitazione/Disabilitazione del Tempo di Scansione Costante, Lettura o Impostazione della Durata del Timer

Per abilitare o disabilitare il modo Sweep Tempo a Costante, per modificare la durata del Tempo di Sweep Costante, per vedere se il Tempo di Sweep Costante è al momento abilitato o per controllarne la durata, SVCREQ 1 può essere compreso nel programma applicativo.

Modi CPU Stop

Il PLC può anche essere in uno dei seguenti due modi Stop:

- Modo Stop con I/O Disabilitati
- Modo Stop con I/O Abilitati

Quando il PLC è nel modo Stop, la CPU non esegue la logica del programma applicativo. Si può configurare la scansione o meno del I/O durante il modo Stop. Nel modo Stop, le comunicazioni con il programmatore e i moduli opzionali intelligenti continuano. Inoltre, nel modo Stop continuano anche il polling delle schede guaste e l'esecuzione della riconfigurazione.

SVCREQ 13 può essere utilizzato nel programma applicativo per fermare il PLC alla fine dello sweep successivo. Tutti gli I/O passeranno al loro stato configurato per default e nella Tabella Guasti PLC verrà inserito un messaggio diagnostico.

Controllo dell’Esecuzione di un Programma

Il set di istruzioni della CPU VersaMax contiene potenti funzioni di controllo che possono essere incluse in un programma applicativo per limitare o modificare il modo in cui la CPU esegue il programma e scansiona gli I/O.

Richiamo di un Blocco di Subroutine

La funzione CALL può essere utilizzata per fare in modo che l’esecuzione di un programma passi a una specifica subroutine. Una logica condizionante posta prima delle funzione Call controlla le circostanze nelle quali la CPU esegue la logica della subroutine. Terminata la subroutine, l’esecuzione del programma riprende dal punto della logica immediatamente successiva all’istruzione CALL.

Creazione di un Fine Logica Temporaneo

La funzione END può essere utilizzata per introdurre un fine logica temporaneo, inseribile ovunque nel programma. Dopo la funzione END non viene più eseguita alcuna logica e l’esecuzione del programma riparte direttamente dall’inizio. Pertanto, la funzione END torna utile per il debug di un programma.

la funzione END non deve essere posta in logiche associate a o richiamate da una struttura di controllo Sequential Function Chart. Se ciò avvenisse, il PLC passerebbe la modo STOP/FAULT al termine dello sweep in corso e verrebbe registrato un errore FC_END.

Esecuzione di parti di Logica senza Flusso Logico di Alimentaz.

Il Relè Master di Controllo può essere utilizzato per eseguire una parte della logica del programma senza flusso logico di alimentazione. Le bobine in tale parte del programma vengono eseguite con flusso di corrente negativa. In un programma, i Relè Master di Controllo possono essere annidati a 8 livelli.

Passaggio a un’Altra Parte del Programma

A differenza del Relè Master di Controllo, che può soltanto far avanzare nel programma, la funzione Jump può spostare l’esecuzione del programma sia avanti che indietro nella logica. Un’altra differenza sta nel fato che quando la funzione Jump è attiva, le bobine nella parte di programma che viene saltata rimangono nelle condizioni precedenti (non eseguite con flusso di corrente negativa, in quanto associate a un Relè Master di Controllo). Le funzioni Jump possono anche essere annidate.

Le funzioni Jump non possono estendersi a blocchi, azioni SFC, transizioni SCF o logiche di pre- post-elaborazione SFC.

Livelli Privilegiati e Password

Il PLC VersaMax fornisce quattro livelli di protezione selezionabili per la sicurezza del sistema. Il primo livello, sempre disponibile, consente solo di leggere i dati PLC, senza possibilità di modifica dell'applicazione. Gli altri tre livelli hanno accesso a ciascun livello protetto da password. Ciascun livello superiore privilegiato permette maggiori possibilità di modifica del/dei livello/i inferiore/i.

Livello	Descrizione
1	Può leggere tutte le memorie dati (%I, %Q, %AQ, %R, ecc.), le tabelle errori e tutti i blocchi programma (dati, valori e costanti). NON può leggere le Password. NON può modificare i valori del PLC. Può cancellare l'intero contenuto di un programmatore.
2	Può scrivere in tutte le memorie dati (%I, %R, ecc.). Può cancellare l'intero contenuto di un programmatore.
3	Può scrivere nel programma applicativo solo nel modo **STOP**. Può cancellare l'intero contenuto di un programmatore.
4	Può leggere e scrivere in tutte le memorie e le password nel modo **RUN** o **STOP.** (Nel modo **RUN** non si possono modificare i dati di configurazione.) Può cancellare l'intero contenuto di un programmatore.

Se non viene impostata una password, il livello di default è il livello 4. Se è stata impostata una password, il livello di default è il livello più elevato non protetto.

Il livello privilegiato e la password possono essere cambiati immettendo dal programmatore un nuovo livello privilegiato e la password. La password esistente per il livello deve essere immessa prima che la CPU permetta la modifica. Un cambio di livello privilegiato dura in effetti per il tempo in cui il collegamento per comunicazioni tra la CPU e il programmatore rimane intatto. In assenza di comunicazioni per 15 minuti, il privilegio torna al livello più alto non protetto.

Le Password

Ai livelli 2, 3 e 4 si può assegnare una password di protezione. Le password vengono immesse mediante il software di programmazione. Ogni password può essere esclusiva o usata per più di un livello.

Cancellazione Totale della Memoria

Dal programmatore è possibile eseguire la cancellazione totale della memoria con la CPU a qualsiasi livello di privilegio. Ciò consente ai tecnici di cancellare la memoria CPU e caricare un nuovo programma applicativo anche se non conoscono la password .

Funzionamento del Commutatore dei Modi Run/Stop

Il commutatore RUN/STOP può essere configurato per avere la CPU nel modo Stop o nel modo Run. Può anche configurato per evitare la scrittura nel programma o nella memoria della configurazione e forzare o evitare dati discreti. Per default, può abilitare i modi Run/Stop e disabilita la protezione della memoria.

Funzionamento dei Modi Run/Stop Configurabili

Se il funzionamento del commutatore dei modi Run/Stop è abilitato, può essere utilizzato per porre la CPU nel modo Run.

- Se la CPU non ha errori fatali e non è nel modo Stop/Fault, portando il commutatore su Run la CPU passa al modo Run. Gli errori NON sono cancellati.
- Se la CPU ha un errore fatale ed è nel modo Stop/Fault, portando il commutatore su Run il LED Run lampeggia per 5 sec. Mentre questo lampeggia, il commutatore della CPU può essere usato per vuotare la tabella errori e porre la CPU nel modo Run. Dopo che il commutatore sarà stato per almeno 1/2 secondo su Run, portarlo su Stop per almeno 1/2 secondo, poi riportarlo su Run. Gli errori vengono cancellati e la CPU passa al modo Run. Il ED smette di lampeggiare e rimane acceso. Se necessario, ripetere questa operazione.
- Se la commutazione non avviene nel modo descritto, il LED Run si spegne, la CPU rimane nel modo Stop/Fault e la tabella errori rimane invariata.

Protezione della Memoria Configurabile

Il commutatore può esser configurato in modo che eviti la scrittura nella memoria programma e di configurazione, nonché la forzatura o l'esclusione di dati discreti.

Sommario del Funzionamento del Commutatore Run/Stop CPU

Configurazione Modo Run/Stop	Configurazione Stop Scans. I/O	Posizione del Commutatore	Funzionamento della CPU
Off	nessun effetto	nessun effetto	Sono ammessi tutti i modi.
On	nessun effetto	Run/On	Sono ammessi tutti i modi.
On	nessun effetto	Stop/Off	la CPU non può passare al modo Run.
Off	nessun effetto	Commutato da Stop a Run	La CPU passa al modo Run in assenza di errori fatali; diversamente il LED Run lampeggia 5 sec.
On	No	Commutato da Run a Stop	Il PLC passa a STOP–NO IO
On	Sì	Commutato da Run a Stop	il PLC passa a STOP–IO

Capitolo
6

Elementi di un Programma Applicativo

Questo capitolo fornisce le informazioni basilari sul programma applicativo.

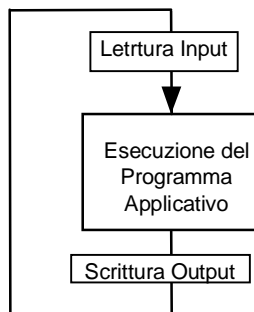
- Struttura di un programma applicativo
- Subroutine
- Linguaggi del programma
- Set di Istruzioni

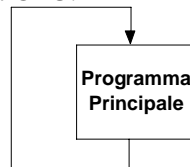
Struttura di Un Programma Applicativo

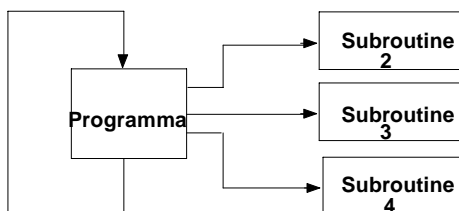
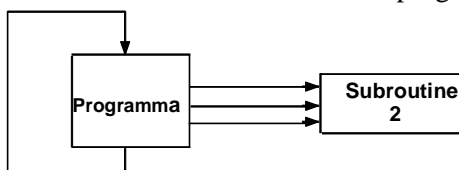
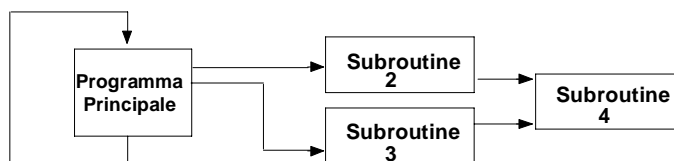
Il programma applicativo è costituito da tutte le logiche necessarie a controllare le operazioni della CPU per PLC e dei moduli del sistema.

I programmi applicativi vengono creati utilizzando il software di programmazione, per poi trasferirli al PLC. I programmi vengono caricati nella memoria non volatile della CPU. Per il caricamento dei programmi applicativi, il PLC dispone di 12 kB di memoria.

Durante lo sweep della CPU (descritto ne precedente capitolo), la CPU legge i dati provenienti dai moduli del sistema e li carica tali dati nelle locazioni configurate nella memoria input. Quindi, la CPU esegue una volta l'intero programma applicativo, utilizzando i dati appena ricevuti. L'esecuzione del programma applicativo crea nuovi dati output, che vengono inviati alle locazioni della memoria configurate per gli output.

Al termine del programma applicativo, la CPU scrive i dati output nei moduli del sistema.

Le Subroutine

Il programma può essere costituito da un programma Principale, che viene eseguito completamente durante lo sweep della CPU.

Oppure, il programma può essere suddiviso in subroutine. La dimensione massima di un programma principale o di un blocco di subroutine è 12kB.

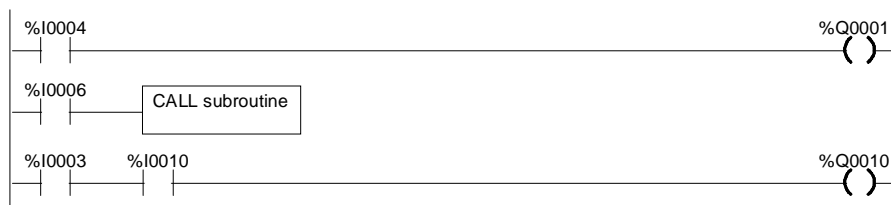
Le subroutine sono semplici da programmare e riducono la quantità totale di logiche. Ogni subroutine può avere il suo nome. Il programma principale potrebbe servire principalmente a sequenziare i blocchi delle subroutine.

Durante l'esecuzione del programma, un blocco di subroutine può essere richiamato varie volte. La logica che deve essere sostituita può essere posta in un blocco di subroutine, riducendo così le dimensioni totali del programma.

Oltre a poter essere richiamate dal programma, le subroutine possono anche essere richiamate da altri blocchi di subroutine. Una subroutine non può mai richiamare se stessa.

Il programma principale è il livello 1, ma può comprendere fino ad altri otto livelli di richiamo annidati.

Dichiarazione di una Subroutine

Una subroutine deve essere dichiarata mediante l'editor di dichiarazione del blocco del software di programmazione.

Richiamo di una Subroutine

Una subroutine viene richiamata nel programma mediante il comando CALL. Per ciascun blocco del programma si possono dichiarare fino a 64 subroutine e fino a 64 comandi CALL.

%I0004 %Q0001
%I0006 CALL subroutine
%I0003 %I0010 %Q0010

Bloccaggio/Sbloccaggio di Subroutine

I blocchi delle subroutine possono essere bloccati o sbloccati mediante il software di programmazione. Vi sono quattro livelli di bloccaggio:

Tipo di Bloccaggio	Descrizione
Vista	Quando bloccato, la subroutine non può essere vista.
Edit	Quando bloccato, non si possono modificare le informazioni contenute nella subroutine.
Vista Perm	La subroutine è permanentemente bloccata e non può essere sbloccata .
Perm Edit	La subroutine è permanentemente bloccata e non può essere sbloccata.

Le subroutine permanentemente bloccate non possono essere sbloccate, anche se il programma viene copiato, salvato in back up o memorizzato. Al contrario, una subroutine che viene bloccata per la vista e l'editazione può essere sbloccata.

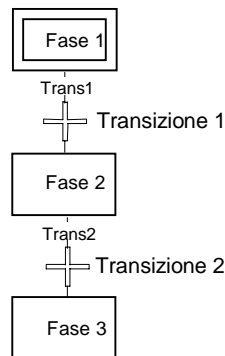
Linguaggi del Programma

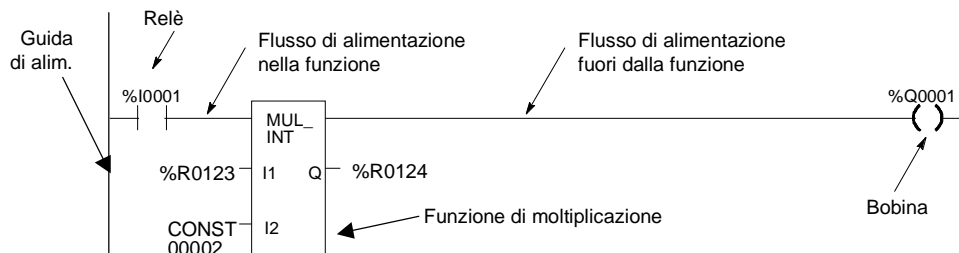
Il linguaggio di base per il programma applicativo è il linguaggio Ladder Diagram. Il programma principale o le subroutine in esso contenute possono anche essere create nel formato Sequential Function Chart. Il software di programmazione può essere usato per creare entrambi questi tipi di logiche.

Sequential Function Chart

Sequential Function Chart (SFC) è un metodo grafico per rappresentare le funzioni di un sistema sequenziale automatizzato come sequenza di fasi e transizioni. Ciascuna fase rappresenta comandi o azioni che possono essere attivi o non attivi.

Il flusso del controllo passa da una fase a quella successiva mediante una transizione condizionale, che può essere vera (1) o falsa (2). Se la condizione della transizione è vera (1), il controllo passa dalla fase corrente (che diventa non attiva) a quella successiva, che diverrà attiva.

La logica associata a una fase viene eseguita quando la fase è attiva. Tale logica viene programmata nel formato Ladder Diagram. Anche le transizioni tra le fasi vengono programmate come logiche Ladder Diagram.

Ladder Diagram

Questo tradizionale linguaggio di programmazione PLC, con struttura del tipo a gradini, viene eseguito dall'alto al basso. L'esecuzione logica viene considerata come "flusso di alimentazione", che procede verso il basso lungo il lato destro di una scala e da sinistra a destra lungo la sequenza di ciascun gradino.

Il flusso dell'alimentazione logica lungo ciascun gradino viene controllato da un set di semplici funzioni del programma che operano come relè meccanici e bobine output. Il fatto che un relè passi o meno il flusso di alimentazione logica lungo un gradino, dipende dal contenuto di una locazione di memoria con la quale il relè è stato associato nel programma. Per esempio, un relè può passare il flusso di alimentazione se è associato a una locazione di memoria che contiene il valore 1. Lo stesso relè non passa il flusso di alimentazione se la locazione di memoria contiene un valore 0.

Se un relè o un'altra funzione in un gradino non passa il flusso logico di alimentazione, il resto del gradino non viene eseguito. L'alimentazione fluirà lungo la guida sinistra al gradino successivo.

All'interno di un gradino vi sono molte funzioni complesse, che possono essere utilizzate per operazioni come il trasferimento di dati caricati in memoria, l'esecuzione di operazioni matematiche e il controllo delle comunicazioni tra la CPU e altri dispositivi del sistema.

Alcune funzioni del sistema, come il salto di una funzione il Relè master di Controllo, possono essere utilizzate per controllare l'esecuzione del programma stesso.

Questo vasto gruppo di relè, bobine e funzione del Ladder Diagram viene definito "Set di istruzioni" della CPU.

Il Set di Istruzioni

La CPU per PLC VersaMax fornisce un potente set di istruzioni per costruire programmi applicativi.

Come guida alle possibilità di programmazione del PLC VersaMax, tutti i relè, le bobine, le funzioni e gli altri elementi del Set di Istruzioni sono riassunti nelle pagine che seguono. Le informazioni complete di riferimento sono comprese nella documentazione e nell'help in linea del software di programmazione.

Contatti

–| |–	**Normalmente aperto**	Passa corrente se il riferimento associato è ON.
–|/|–	**Normalmente chiuso**	Passa corrente se il riferimento associato è OFF.
<+>——	**Continuazione**	Passa corrente sulla destra se la precedente bobina di continuazione è impostata su ON.

Bobine

–( )–	**Normalmente aperto**	Imposta su ON il riferimento associato se la bobina riceve corrente. Altrimenti è OFF.
–(/)–	**Negato**	Imposta su ON il riferimento associato se la bobina non riceve corrente. Altrimenti è OFF.
–(↑)–	**Tensione Positiva**	Se il flusso di corrente era OFF su questa bobina l'ultima volta che è stata eseguita e ora è ON, la bobina viene attivata, diversamente passa a OFF.
–(↓)–	**Tensione negativa**	Se il flusso di corrente era ON su questa bobina l'ultima volta che è stata eseguita e ora è OFF, la bobina viene attivata, diversamente passa a OFF.
–(S)–	**SET**	Imposta su ON i riferimenti discreti associati se la bobina riceve corrente. Rimane impostata fino a che verrà resettata da un –(R)– bobina.
–(R)–	**RESET**	Imposta su OFF i riferimenti discreti associati se la bobina riceve corrente. Rimane resettata fino a che verrà impostata da un –(S)– bobina.
–(SM)–	**SET Retentivo**	Imposta su ON i riferimenti associati se la bobina riceve corrente. Il riferimento rimane impostato fino a che verrà resettato da un –(RM)– bobina. Il suo stato viene mantenuto anche in caso manchi corrente o di una transizione **STOP-TO-RUN**.
–(RM)–	**RESET Retentivo**	Resetta su OFF i riferimenti associati discreti se la bobina riceve corrente. Il riferimento rimane resettato fino a che verrà impostato da un –(RM)– bobina. Il suo stato viene mantenuto anche in caso manchi corrente o di una transizione **STOP-TO-RUN**.
–(/M)–	**ritenzione Negata**	Imposta su ON i riferimenti discreti associati se la bobina non riceve corrente. Il suo stato viene mantenuto anche in caso manchi corrente o di una transizione **STOP-TO-RUN**. Diversamente sarà OFF
–(M)–	**Ritenzione**	Imposta su ON i riferimenti discreti associati se la bobina riceve corrente. Il suo stato viene mantenuto anche in caso manchi corrente o di una transizione **STOP-TO-RUN**. Diversamente sarà OFF.
——<+>	**Continuazion e**	Se la corrente alla bobina è ON, la bobina di continuazione imposta su ON il successivo contatto di continuazione. Se manca corrente, la bobina di continuazione imposta il successivo contatto su OFF.

Timer e Contatori

ondtr	**Timer Stopwatch Tempo On**	Accumula il tempo durante il quale riceve corrente. Il valore della corrente viene azzerato quando il segnale Reset riceve tensione.
oftd	**Timer Tempo Off**	Accumula in tempo durante il quale NON riceve corrente.
tmr	**Timer Tempo ON**	Accumula il tempo durante il quale riceve corrente. Il valore della corrente viene azzerato quando non c'è flusso di corrente.
upctr	**Contatore Su**	Incrementa di 1 ogni volta che la funzione riceve una corrente transitoria.
dnctr	**Contatore Giù**	Conta alla rovescia rispetto a un valore preimpostato ogni volta che la funzione riceve una corrente transitoria.

Funzioni Matematiche

add	**Addizione**	Somma due numeri.
sub	**Sottrazione**	Sottrae un numero da un altro.
mul	**Moltiplicazione**	Moltiplica due numeri.
div	**Divisione**	Divide un numero per un altro, dando il quoziente.
mod	**Divisione Modulare**	Divide un numero per un altro, dando il resto.
expt	**Potenza di X**	Porta X alla potenza specificata da IN e pone il risultato in Q.
sin	**Seno Trigonometrico**	Trova il seno trigonometrico di un numero reale.
cos	**Coseno Trigonometrico**	Trova il coseno trigonometrico di un numero reale.
tan	**Tangente Trigonometrica**	Trova la tangente trigonometrica di un numero reale.
asin	**Seno Inverso**	Trova il seno inverso di un numero reale.
acos	**Coseno Inverso**	Trova il coseno inverso di un numero reale.
atan	**Tangente Inversa**	Trova la tangente inversa di un numero reale.
deg	**Conversione in Gradi**	Esegue la conversione da RADIALE in GRADI da un valore radiale reale.
rad	**Conversione in Radiante**	Esegue la conversione da GRADI in RADIALE da un valore in gradi reale.
sqroot	**Radice Quadrata**	Trova la radice quadrata di un integrale e di un valore reale.
Log	**Logaritmo in base 10**	Trova il logaritmo in base 10 di un valore reale.
ln	**Logaritmo naturale**	Trova il logaritmo naturale di un numero reale.
exp	**Potenza di e**	Porta la base del logaritmo naturale alla potenza specificata dall'input.

Funzioni Relazionali

eq	**Uguale**	Controlla l'uguaglianza tra due numeri.
ne	**Non Uguale**	Controlla la non uguaglianza tra due numeri.
gt	**Maggiore Di**	Controlla se un numero è superiore a un altro. Passa corrente se il primo è superiore al secondo.
ge	**Maggiore di o Uguale a**	Controlla se un numero è superiore o uguale a un altro.
lt	**Minore di**	Controlla se un numero è inferiore a un altro.
le	**Minore di o Uguale a**	Controlla se un numero è inferiore o uguale a un altro.
range	**Gamma**	Controlla il valore input rispetto a un range di due numeri.

Funzioni Bit

and	**Logica AND**	Esegue la logica AND di due stringhe bit.
or	**Logica OR**	Esegue la logica OR di due stringhe bit.
xor	**Logica Esclusiva OR**	Esegue la logica esclusiva OR di due stringhe bit.
not	**Inversione Logica**	Esegue un'inversione logica di una stringa bit.
shl	**Sposta a Sinistra**	Sposta a sinistra una stringa bit.
shr	**Sposta a Destra**	Sposta a destra una stringa bit.
rol	**Ruota a Sinistra**	Ruota a sinistra una stringa bit.
ror	**Ruota a Destra**	Ruota a destra una stringa bit.
bittst	**Bit Test**	Testa un bit all'interno di una stringa bit.
bitset	**Bit Set**	Imposta come vero un bit all'interno di una stringa.
bitclr	**Cancella Bit**	Imposta come falso un bit all'interno di una stringa.
bitpos	**Posizione Bit**	Localizza un bit impostato come vero in una stringa bit.
mskcmp	**Comparazione Mascherata**	Esegue una comparazione mascherata di due array.

Funzioni Spostamento Dati

move	**Sposta**	Sposta uno o più bit di dati.
blkmov	**Sposta Blocco**	Sposta un blocco di fino a 7 costanti.
blkclr	**Annulla Blocco**	Azzera uno o più bit/word della memoria.
shfreg	**Sposta Registro**	Sposta uno o più bit di dati o word lungo un blocco di memoria
bitseq	**Sequenziatore Bit**	Sequenzia un 1 lungo un gruppo di bit nella memoria PLC.
comreq	**Richiesta di Comunicazione**	Invia una richiesta di comunicazione.

Funzioni Tabella

arrmov	**Sposta Array**	Copia un numero specificato di elementi dati da un array sorgente a un array di destinazione.
srh eq	**Cerca Uguale**	Cerca array di valore uguale a un valore specificato.
srh ne	**Cerca non Uguale**	Cerca array di valore non uguale a un valore specificato.
srh gt	**Cerca Maggiore Di**	Cerca array di valore superiore a un valore specificato.
srh ge	**Cerca maggiore Di o Uguale**	Cerca array di valore superiore o uguale a un valore specificato.
srh lt	**Cerca Meno Di**	Cerca array di valore inferiore a un valore specificato.
srh le	**Cerca mano Di o Uguale**	Cerca array di valore inferiore o uguale a un valore specificato.

Funzioni di Conversione

→bcd4	**Converti in BCD4 (Da INT)**	Converte un numero di 4 cifre in formato BCD.
→word	**Converti in Word (Da REALE)**	Converte un valore reale in formato word.
→int	**Converti in INT (Da BCD4 o REALE)**	Converte un numero nel formato intero con segno.
→tdint	**Converti in DINT (Da BCD4 o REALE)**	Converte un numero nel formato integrale doppia precisione.
→real	**Converti in Reale (Da INT, DINT, BCD4 o WORD)**	Converte un numero nel formato valore reale.
→→int	**Tronca a INT (Da REALE)**	Tronca a 16-bit un numero con segno. Il range è da -32,768 a +32,767.
→→dint	**Tronca a Doppia Precisione INT (Da REALE)**	Tronca a 32-bit un numero con segno. Il range è da -2,147,483,648 a +2,147,483,647.

Funzioni di Controllo

call	**Richiama**	Passa l’esecuzione di un programma a un blocco di subroutine specificato.
do io	**Esegui I/O**	Esegue immediatamente una gamma specificata di input o output (tutti gli input o gli output di un modulo verranno eseguiti se su tale modulo nessun indirizzo è incluso nella funzione – non vengono eseguiti aggiornamenti parziali I/O sul modulo)
pidind	**Algoritmo PID Indipendente**	Seleziona l’algoritmo PID indipendente non interattivo.
pidisa	**Algoritmo ISA PID**	Seleziona l’algoritmo PID ISA.
end	**Fine Temporanea della Logica**	Il programma esegue dal primo gradino all’ultimo o fino all’istruzione END (fine) se questa precede l’ultimo gradino. Questa istruzione o utile per il debug.
commnt	**Commento**	Spiegazione di un gradino.
svcreq	**Service Request**	Speciale funzione service del PLC.
mcr	**Relè Master di Controllo**	Avvia la gamma di un relè master di controllo. Un MRC fa eseguire tutti i gradini tra l’MRC e il successivo ENDMRC senza passaggio di corrente. Si possono annidare fino a 8 MRC.
endmcr	**Fine Relè Master di Controllo**	Fine della gamma del relè master di controllo.
jump	**Salto**	Salta una posizione specificata indicata da una LABEL (etichetta) nella logica.
label	**Label**	La posizione bersaglio di un’istruzione Salto (jump). Alla stessa etichetta possono far riferimento varie istruzioni jump.

Capitolo 7

Dati del Programma

Questo capitolo descrive i tipi di dati che possono essere utilizzati nel programma applicativo e spiega come questi dati vengono caricati nella memoria PLC.

- Riferimenti alla memoria dati
- Ritenzione dei dati
- Utilizzo di nomi abbreviati e di descrizioni per riferimenti al programma
- Riferimenti alle condizioni del sistema
- Contatti di temporizzazione
- Come programmare le funzioni e gestire i dati numerici

Riferimenti alla Memoria Dati

Il PLC carica i dati del programma in entrambe le memorie: bit e word. Queste due memorie sono entrambe divise in due tipi diversi con caratteristiche specifiche.

Allocazione in Memoria

La seguente tabella indica i tipi e le dimensioni di memoria disponibili.

Memoria Programma	12288 byte
Registri (%R)	2048 word
Input Discreti (%I)	2048 punti
Output Discreti (%Q)	2048 punti
Input Analogici (%AI)	128 word
Output Analogici (%AQ)	128 word
Interni Discreti (%M)	1024 punti
Temporanei Discreti (%T)	256 punti
Global Genius Discreto (%G)	1280 punti

Come spiegato oltre, per convenzione ciascun tipo viene normalmente usato per uno specifico tipo di dati. L'assegnazione in memoria è però molto flessibile.

Le singole locazioni di memoria vengono indicizzate con identificatori alfanumerici, detti riferimenti. La lettera del prefisso del riferimento identifica l'area di memoria. Il valore numerico è l'offset all'interno della memoria.

Riferimenti della Memoria Word

Ogni indirizzo (riferimento) della memoria word ha un limite di 16-bit. Il PLC, per i dati caricati nella memoria word, usa tre tipi di riferimenti.

- **%AI** Normalmente usato per input analogici.
- **%AQ** Normalmente usato per output analogici .
- **%R** I registri sono di solito usati per caricare dati prog. in formato word.

La memoria word è riportata qui sotto con un esempio che indica dieci indirizzi; ciascuno con 16 bit che assieme contengono un valore. Nella memoria word, il PLC non può accedere a bit individuali.

indirizzi	
1	12467
2	12004
3	231
4	359
5	14
6	882
7	24
8	771
9	735
10	0000

Riferimenti della memoria Bit

Ogni indirizzo (riferimento) della memoria bit ha un limite di bit. I dati vengono caricati nella memoria bit come nell’esempio. La figura mostra 160 bit indirizzati individualmente, con l’indirizzo 1 in alto a sinistra e il 160 in basso a destra

addresses

1	2	3	4	5	6	7	8								
0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0
1	1	1	1	0	0	0	1	1	0	0	1	0	0	0	0
1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	0
0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	0
1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	0
1	1	0	1	0	0	0	1	1	1	0	1	0	0	0	0
1	1	0	0	0	0	0	1	1	0	1	1	1	0	1	1
1	0	0	1	0	0	0	1	1	0	1	1	1	0	0	1
0	0	0	1	0	0	0	0	1	0	1	0	1	0	0	1

... 160

Per i dati caricati in memoria, il PLC utilizza sei tipi di riferimenti.

%I	Normalmente usato per input discreti e visibile nella tabella Input Status.
%Q	Normalmente usato per output fisici di riferimento e visibile nella Tabella Output Status. Un riferimento %Q può essere di ritenzione o meno, in funzione del suo utilizzo nel programma.
%M	Normalmente utilizzato per rappresentare riferimenti interni. Un riferimento %M specifico può essere di ritenzione o meno, in funzione del suo utilizzo nel programma.
%T	Usato per riferimenti temporanei, che possono essere utilizzati molte volte in un programma. I dati con il riferimento %T non sono ritenuti se manca corrente o in caso di transizioni RUN-TO-STOP-TO-RAUN. I riferimenti %T non possono essere usati con bobine ritentive.
%S	Riferimenti alle condizioni del sistema, che hanno predefinizioni specifiche. ■ %S, %SA, %SB, e %SC possono essere usati per ogni tipo di contatto logico. ■ %SA, %SB, e %SC possono essere usati con bobine ritentive. ■ %S può essere usato come input in funzioni o blocchi di funzioni. ■ %SA, %SB, e %SC possono essere utilizzati come input od output di funzioni o di blocchi di funzioni.
%G	Usato per Dati Globali. I dati nel riferimento %G vengono ritenuti anche se manca corrente. I riferimenti %G possono essere usati con contatti e bobine ritentive, ma non con bobine non ritentive.

Bit di Transizione e Bit Override

I riferimenti %I, %Q, %M, e %G hanno associati bit di transizioni e di override.

I riferimenti %T, %S, %SA, %SB, e %SC hanno associati solo bit di transizione.

La CPU usa bit di transizione per bobine di transizione. Se si impostano bit di override, i riferimenti associati possono essere cambiati solo dal programmatore.

Ritenzione dei Dati

Un dato è ritentivo se viene salvato automaticamente quando si spegne la CPU o viene tolta corrente. I seguenti dati sono ritentivi:

- Logica del programma
- Tabella errori e diagnostica
- Gli override
- I dati word (%R, %AI, %AQ)
- I dati Bit (i bit di errore %I, %SC, %G, sono bit riservati)
- Dati word caricati in %Q e %M.
- Dati nei riferimenti %Q o %M che vengono usati come output di un blocco di funzione o con bobine ritentive:

  -(M)- bobine ritentive
  -(/M)- bobine ritentive negate
  -(SM)- bobine ritentive SET
  -(RM)- bobine ritentive RESET

  L'ultima volta che un riferimento %Q o %M viene usato con un bobina, il tipo di bobina determina se il dato è ritentivo o non ritentivo. Per esempio, se %Q0001 era l'ultimo programmato come riferimento di una bobina ritentiva, il dato %Q0001 è ritentivo. Ma se %Q0001 era l'ultimo programmato su una bobina non ritentiva, il dato %Q0001 è non ritentivo.

  - I riferimenti %Q o %M sono stati resi ritentivi dichiarando specificamente che lo dovevano essere. I riferimenti %Q e %M per default passano a non ritentivi.

I seguenti dati sono non ritentivi:

- Le condizioni delle bobine di transizione.
- I dati %T
- I dati %S, %SA, e %SB (ma il dato bit %SC È ritentivo).
- I riferimenti %Q e %M che non sia stato dichiarato essere ritentivi
- I riferimenti %Q e %M utilizzati con bobine non ritentive:

  -( )- bobine
  -(/)- bobine negate
  -(S)- bobine SET
  -(R)- bobine RESET

Uso di Nomi Abbreviati e Descrizioni per Riferimenti

Molti programmi usano come riferimenti nomi abbreviati descrittivi per rendere la logica del programma più facile da capire quando viene letta.

Nomi Abbreviati

Un nome abbreviato può contenere da 1 a 7 caratteri, che possono essere dalla A alla Z e da 0 a 9 o caratteri speciali come +, -, _, %, #, @, <, >, =, e &. Il primo carattere deve essere una lettera. A ogni riferimento del programma può essere assegnato un nome abbreviato. I nomi abbreviati, il cui uso è opzionale, vengono assegnati nella Tabella Dichiarazione Variabili.

Un nome abbreviato non risente della maiuscole o minuscole, a meno che non ne venga assegnato uno a una subroutine. In tal caso, l'uso locale del nome abbreviato è in maiuscole e l'uso globale e in minuscole.

Nomi Abbreviati Locali in Subroutine

Ogni blocco di subroutine può avere il suo uso locale di nomi abbreviati. Lo stesso riferimento può avere vari nomi abbreviati locali in diversi blocchi di subroutine:

BLOCCO A %R0001 Light_1
BLOCCO B %R0001 Light_2

Due blocchi di subroutine possono avere lo stesso nome abbreviato per riferimenti diversi, se dichiarato come variabile locale:

BLOCCO A %R0001 RESET
BLOCCO B %R0002 RESET

Descrizione dei Riferimenti

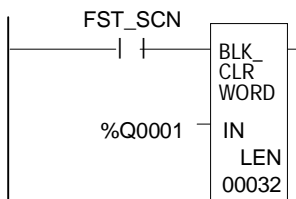
La Descrizione di un Riferimento è un testo opzionale costituito da un massimo di 32 caratteri e può essere associata a riferimenti della macchina o a identificatori impliciti (es. nome del programma, blocco di subroutine, oppure JUMP/LABEL/MCR/ENDMCR).

Una descrizione di un riferimento può essere usata con o senza nome abbreviato.

Riferimenti alle Condizioni del Sistema

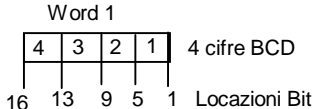
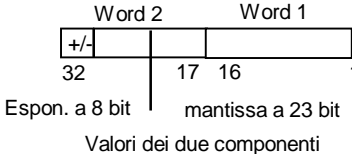
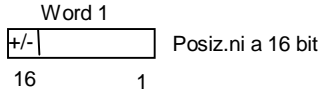
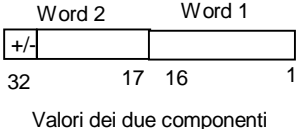
Il PLC carica i dati delle condizioni del sistema nelle memorie %S, %SA, %SB, e %SC. Ciascun riferimento delle condizioni del sistema ha un nome abbreviato di descrizione. Per esempio, i riferimenti “time tick” vengono chiamati T_10MS, T_100MS, T_SEC, e T_MIN. Esempi convenienti di riferimenti comprendono FST_SCN, ALW_ON, e ALW_OFF.

Uso dei Riferimenti delle Condizioni del Sistema

I riferimenti delle condizioni del sistema possono essere usati secondo le necessità nei programmi applicativi. Per esempio, il seguente blocco di funzione usa il riferimento delle condizioni FST_SCN (prima scansione) per controllare il flusso di corrente alla funzione Cancella Blocco. In questo esempio, all’accensione, 32 word della memoria %Q (512 punti) che cominciano da %Q0001 vengono riempite di zeri.

Riferimenti %S

I riferimenti nella memoria %S sono di sola lettura.

Riferim.	Nome Abb.	Definizione
%S0001	FST_SCN	Impostato su 1 quanto lo sweep in corso è il primo.
%S0002	LST_SCN	Resettato da 1 a 0 quando lo sweep in corso è l’ultimo.
%S0003	T_10MS	Contatto timer da 0,.01 secondi
%S0004	T_100MS	Contatto timer da 0,1 secondi.
%S0005	T_SEC	Contatto timer da 1,0 secondi.
%S0006	T_MIN	Contatto timer da 1,0 minuti.
%S0007	ALW_ON	Sempre ON.
%S0008	ALW_OFF	Sempre OFF.
%S0009	SY_FULL	Impostato quando la tabella errori PLC è completa. Cancellato quando si rimuove un’immissione e quando la tabella errori PLC viene cancellata.
%S0010	IO_FULL	Impostato quando la tabella errori di I/O è completa. Cancellato quando si rimuove un’immissione dalla tabella errori I/O o quando si cancella la tabella errori I/O.
%S0011	OVR_PRE	Impostato quando esiste un override nella memoria %I, %Q, %M, o %G.
%S0012		riservato
%S0013	PRG_CHK	Impostato nel controllo in background del programma.
%S0014	PLC_BAT	Impostato per indicare batteria CPU scarica. Il riferimento del contatto viene aggiornato a ogni sweep.
%S0015, 16		riservato
%S0017	SNPXACT	SNP-X host è attivamente connesso alla porta 1 CPU. (La porta 2 si disabilita per default e deve essere attivata con un CRQ).
%S0018	SNPX_RD	SNP-X host ha letto dati dalla porta 1 CPU.
%S0019	SNPX_WT	SNP-X host ha scritto dati nella porta 1 CPU.
%S0020		Impostato ON quando una funzione relazionale che usa dati REALI viene eseguita con successo. Annullato quando uno degli input è NaN (non un numero).
%S0021	FF_OVR	Impostato per riferire un override di un Errore fatale.
%S0022	USR_SW	Impostato per riflettere lo stato del nodo switch della CPU 1 = Run/On 0 = Stop/Off
%S0023-32		riservato

Riferimenti %SA, %SB, e %SC

I riferimenti nelle memorie %SA, %SB, e %SC possono essere letti e scritti.

Riferim.	Nome Abb.	Definizione
%SA0001	PB_SUM	Impostato quando una checksum calcolata dal programma applicativo non corrisponde a quella di riferimento. Se l’errore era dovuto a un guasto temporaneo, il bit discreto può essere cancellato ricaricando il programma sulla CPU. Se era dovuto a un guasto hard della RAM, la CPU va sostituita.
%SA0002	OV_SWP	Impostato quando un PLC nel modo SWEEP COSTANTE rileva che lo sweep precedente è durato più del tempo specificato. Cancellato quando il PLC rileva che lo sweep precedente non è durato più del tempo specificato. Cancellato anche durante la transizione da STOP a RUN.
%SA0003	APL_FLT	Impostato in caso di errore applicativo. Cancellato quando il PLC passa dal modo STOP al modo RUN.
%SA0004-8		riservato
%SA0009	CFG_MM	Impostato quando viene rivelata un’incongruenza di configurazione all’avviamento o nel caricamento. Cancellato all’accensione del PLC dopo aver corretto le condizioni.
%SA0010	HRD_CPU	Impostato quando la diagnostica scopre un problema hardware della CPU. Cancellato sostituendo il modulo CPU.
%SA0011	LOW_BAT	Impostato se la batteria è scarica. Cancellato sostituendo la batteria e riavviando il PLC.
%SA0012,13		riservato
%SA0014	LOS_IOM	Impostato quando un modulo di I/O non comunica con la CPU. Cancellato sostituendo il modulo e riaccendendo il sistema.
%SA0015	LOS_SIO	Impostato quando un modulo opzionale non comunica con la CPU. Cancellato sostituendo il modulo e riaccendendo il rack principale.
%SA0016-18		riservato
%SA0019	ADD_IOM	Impostato quando viene aggiunto un modulo di I/O. Cancellato riaccendendo il PLC e quando la configurazione si abbina al sistema dopo un caricamento.
%SA0020	ADD_SIO	Impostato quando viene aggiunto un modulo di I/O. Cancellato riaccendendo il PLC e quando la configurazione si abbina al sistema dopo un caricamento.
%SA0021-26		riservato
%SA0027	HRD_SIO	Impostato in caso di errore hardware in un modulo opzionale. Cancellato sostituendo il modulo e riaccendendo il PLC.
%SA0028-30		riservato
%SA0031	SFT_SIO	Impostato in caso di errore software irreversibile in un modulo opzionale. Cancellato riaccendendo il PLC e quando la configurazione si abbina all’hardware.

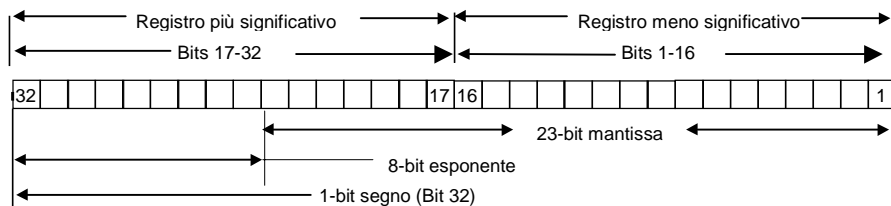
Riferim.	Nome Abb.	Definizione
%SB0001-9		riservato
%SB0010	BAD_RAM	Impostato quando la CPU rileva una memoria RAM guasta all’accensione. Cancellato quando la memoria RAM è valida all’accensione.
%SB0011	BAD_PWD	Impostato in caso di password di accesso errata. Cancellato quando viene cancellata la tabella errori PLC.
%SB0012		riservato
%SB0013	SFT_CPU	Impostato in caso di errore software irreversibile. Cancellato cancellando la tabella errori PLC.
%SB0014	STOR_ER	Impostato in caso di errore durante il caricamento del programma. Cancellato quando il caricamento avviene con successo.
%SC0001-8		riservato
%SC0009	ANY_FLT	Impostato in caso di qualsiasi errore. Cancellato quando entrambe le tabelle errori non hanno immissioni.
%SC0010	SY_FLT	Impostato in caso di qualsiasi errore che iscrive un errore nella tabella errori PLC. Cancellato quando la tabella errori PLC non ha immissioni.
%SC0011	IO_FLT	Impostato in caso di qualsiasi errore che iscrive un errore nella tabella errori di I/O. Cancellato quando la tabella errori di I/O non ha immissioni.
%SC0012	SY_PRES	Impostato fino a quando c’è almeno un’immissione nella tabella errori PLC. Cancellato quando la tabella errori PLC non ha immissioni.
%SC0013	IO_PRES	Impostato fino a quando c’è almeno un’immissione nella tabella errori di I/O. Cancellato quando la tabella errori di I/O non ha immissioni.
%SC0014	HRD_FLT	Impostato in caso di errore hardware. Cancellato quando entrambe le tabelle errori non hanno immissioni.
%SC0015	SFT_FLT	Impostato in caso di errore software. Cancellato quando entrambe le tabelle errori non hanno immissioni.

Come Programmare Funzioni e Gestire Dati Numerici

Indipendentemente da dove i dati sono caricati in memoria – in una memoria bit o in una memoria word – il programma applicativo li può gestire come tipi di dati diversi.

Tipo	Nome	Descrizione	Formato dati
BIT	Bit	Un dato Bit è la più piccola unità di memoria e ha due stati: 1 o 0.	
BYTE	Byte	Un dato tipo Byte ha un valore di 8 bit. Il range valido è da 0 a 255 (da 0 a FF in esadecimali).	
WORD		Un dato tipo word usa una memoria dati di 16 bit consecutivi, ma i bit nella locazione dati, invece di rappresentare un numero, sono bit indipendenti uno dall'altro. Ciascun bit rappresenta il proprio stato binario (1 o 0). Il range valido per i valori word va da 0 a +65,535 (FFFF).	Word 1 Posiz.ni a 16 bit 16 1
BCD-4	Decimale di quattro cifre in Codice Binario	I numeri di quattro cifre BCD usano locazioni di memoria per dati a 16 bit. Ciascuna cifra BCD usa quattro bit e piò rappresentare numeri da 0 a 9. La codifica BCD di 16 bit ha un range di valori da 0 a 999.	Word 1 4 3 2 1 4 cifre BCD 16 13 9 5 1 Locazioni Bit
REAL	Virgola mobile	I numeri reali usano due locazioni di memoria consecutive a 16 bit. Il range di numeri che può essere caricato in questo formato va da ±1.401298E-45 a ± 3.402823E-38. Ulteriori informazioni alla pag. successiva.	Word 2 Word 1 +/- 32 17 16 1 Espon. a 8 bit mantissa a 23 bit Valori dei due componenti
INT	Integrale con segno	I dati Integrali con segno usano locazioni di memoria a 16 bit. Gli Integrali con Segno sono rappresentati in complementi di notazioni di 2. Il bit 16 è il bit del segno. (0 = positivo, 1 = negativo). Il loro range va da -32,768 a +32,767.	Word 1 +/- Posiz.ni a 16 bit 16 1
DINT	Integrale con segno di Doppia Precisione	I dati Integrali con Segno di doppia precisione usano due locazioni consecutive di memoria a 16 bit. Sono rappresentati in complementi di notazione di 2. Il Bit 32 è il bit del segno, (0 = positivo, 1 = negativo). Il loro range va da -2,147,483,648 a +2,147,483,867.	Word 2 Word 1 +/- 32 17 16 1 Valori dei due componenti

Numeri Reali

I dati REALI, che possono essere usati per certe funzioni matematiche e numeriche, sono in effetti dati a virgola mobile. I numeri a virgola mobile vengono caricati nel formato di precisione singola IEEE-standard. Questo formato richiede 32 bit, che occupano due word PLC adiacenti a 16 bit.

Per esempio, se il numero a virgola mobile occupa i registri %R0005 e %R0006, %R0005 sarà il registro meno significativo e %R0006 il registro più significativo.

La gamma di numeri che possono essere caricati in questo formato va da ± 1.401298E–45 a ± 3.402823E+38 e il numero zero.

Errori in Numeri Reali e in Operazioni

Quando una funzione REALE genera un numero superiore a 3.402823E+38 o inferiore a -3.402823E+38 si ha un overflow. L'OK per l'output della funzione passa a OFF e il risultato viene impostato come infinito positivo (per un numero superiore a 3.402823E+38) o come infinito negativo (per un numero inferiore a –3.402823E+38). Si potrà determinare se ciò è avvenuto testando il senso dell'OK output.

POS_INF	= 7F800000h	– IEEE Rapp. infinito positivo in esadecimali.
NEG_INF	= FF800000h	– IEEE Rapp. infinito negativo in esadecimali.

Se gli infiniti prodotti da un overflow vengono usati come operandi per funzioni REALI, possono causare un risultato indefinito, che viene riportato come NaN (non un numero). Per esempio, la somma tra un infinito positivo e un infinito negativo è indefinita. Quando viene richiamata la funzione ADD_REAL e gli operandi sono un infinito positivo e un infinito negativo, il risultato sarà NaN.

Quando un risultato NaN viene inviato a un'altra funzione, NaN passa nel risultato. Per esempio, se NaN_ADD è il primo operando per la funzione SUB_REAL, il risultato di SUB_REAL sarà NaN_ADD. Se entrambi gli operandi di una funzione sono NaN, il primo passerà oltre. Dato che questa funzione propaga i NaN nelle funzioni, si può identificare la funzione dove NaN è stato originato.

Ogni funzione capace di generare NaN dispone di un NaN specialistico che lo identifica.

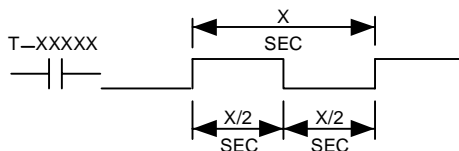
NaN_ADD.	= 7F81FFFFh	– Errore di addizione reale in esadec.
NaN_SUB	= 7F81FFFFh	– Errore di sottrazione reale in esadec.
NaN_MUL	= 7F82FFFFh	– Errore di moltiplicaz. reale in esadec.
NaN_DIV	= 7F83FFFFh	– Errore di divisione reale in esadec.
NaN_SQRT	= 7F84FFFFh	– Err.di radice quadrata reale in esadec.
NaN_LOG	= 7F85FFFFh	– Errore di logaritmo reale in esadec.
NaN_POW0	= 7F86FFFFh	– Errore di esponente reale in esadec.
NaN_SIN	= 7F87FFFFh	– Errore di seno reale in esadec.
NaN_COS	= 7F88FFFFh	– Errore di coseno reale in esadec.
NaN_TAN	= 7F89FFFFh	– Errore di tangente reale in esadec.
NaN_ASIN	= 7F8AFFFFh	– Errore di seno inverso reale in esadec.
NaN_ACOS	= 7F8BFFFFh	– Err. di coseno inverso reale in esadec.
NaN_BCD	= 7F8CFFFFh	– Errore BCD-4 rispetto al reale.
REAL_INDEF	= FFC00000h	– Reale indefinito, err. di divisione per 0.

Contatti di Temporizzazione

Vi sono quattro contatti di temporizzazione, che possono essere usati per fornire impulsi regolari per il flusso di corrente ad altre funzioni del programma. I quattro contatti di temporizzazione hanno durata di: 0,01 secondi, 0,1 secondi, 1,0 secondo e 1 minuto.

Lo stato di questi contatti non cambia durante l'esecuzione dello sweep. Questi contatti forniscono impulsi di durata uguale ON e OFF.

I riferimenti di questi contati sono T_10MS (0.01 secondi), T_100MS (0.1 secondi), T_SEC (1.0 secondo), e T_MIN (1 minuto).

Il seguente schema di temporizzazione rappresenta la durata on/off di questi contatti.

Questi contati di temporizzazione rappresentano locazioni specifiche nella memoria %S.

Capitolo 8

La Funzione Service Request

Questo capitolo spiega la funzione Service Request (SVCREQ), che richiede interventi speciali sul PLC. Vengono descritti i parametri SVCREQ per la CPU VersaMax.

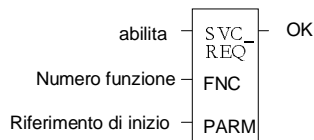
- Numeri della Funzione SVCREQ
- Formato della funzione SVCREQ
- SVCREQ 1: Modifica/Leggi il Timer Sweep Costante
- SVCREQ 2: Leggi la Finestra Tempi
- SVCREQ 3: Modifica Modo Finestra Comunicazioni Programmatore
- SVCREQ 4: Modifica Modo Finestra Comunicazioni Sistema
- SVCREQ 6: Cambia/Leggi il Numero di Word per la Checksum
- SVCREQ 7: Lettura o Modifica data e ora
- SVCREQ 8: Reset del Timer Watchdog
- SVCREQ 9: Lettura Tempo Sweep dall'Inizio dello Sweep
- SVCREQ 10: Lettura Nome Folder
- SVCREQ 11: Lettura ID PLC
- SVCREQ 13: Spegnimento (Arresto) del PLC
- SVCREQ 14: Cancella Errori
- SVCREQ 15: Lettura Ultimo Errore Registrato
- SVCREQ 16: Lettura Tempo Trascorso
- SVCREQ 18: Lettura Stato Override I/O
- SVCREQ 23: Lettura Master Checksum
- SVCREQ 26/30: Interrogazione I/O

Numeri della Funzione SVCREQ

Ciascuna Service Request ha il proprio numero di funzione, come riportato nella tabella qui sotto.

Funzione #	Descrizione
1	Cambia/Leggi il Timer Sweep Costante
2	Lettura Finestra Tempi
3	Cambio Modo e Tempo Finestra Comunicazioni Programmatore
4	Cambio Modo e Tempo Finestra Comunicazioni Sistema
5	riservato
6	Cambia/Leggi il Numero di Word per la Checksum
7	Cambia/Leggi Data e Ora
8	Reset del Timer Watchdog
9	Lettura Tempo Sweep dall'Inizio dello Sweep
10	Lettura Nome Folder
11	Lettura ID PLC
12	riservato
13	Spegnimento del PLC
14	Cancella Tabella Errori
15	Lettura Ultimo Errore Registrato
16	Lettura Tempo Trascorso
17	riservato
18	Lettura Stato di Override I/O
19-22	riservato
23	Lettura Master Checksum
26/30	Interrogazione I/O
27, 28	riservato
29	Lettura Tempo Strumento spento
31-255	riservato

Formato della Funzione SVCREQ

La funzione SVCREQ ha tre input e due output.

Quando SVCREQ riceve corrente, al PLC viene richiesto di eseguire il numero di funzione FNC indicato. I parametri della funzione si trovano all'inizio del riferimento dato per PARM. Questo è l'inizio del "blocco parametro" della funzione. Il numero del riferimento a 16 bit richiesto dipende dalla funzione SVCREQ in uso.

I blocchi parametro possono essere usati sia come input per la funzione che per la locazione dove i dati possono essere inviati dopo l'esecuzione della funzione. Pertanto, ai dati di ritorno dalla funzione si accede nella stessa locazione specificata da PARM.

La funzione SVCREQ fa passare il flusso di corrente, a meno che non siano specificati un numero di funzione errato, parametri non corretti o riferimenti fuori range. Specifiche funzioni SVCREQ hanno ulteriori cause di errore.

Parametri delle Funzione SVCREQ

Input/ Output	Scelte	Descrizione
abilita	flusso	Quando è attivato abilita, viene eseguito il servizio richiesto.
FNC	costanti I, Q M, T, G, R, AI, AQ,	Contiene la costante o il riferimento per il servizio richiesto.
PARM	I, Q M, T, G, R, AI, AQ	Contiene il riferimento di inizio per il blocco parametro del servizio richiesto
ok	nessun flusso	OK viene attivato se la funzione termina senza errori.

Esempio della Funzione SVCREQ

Nell'esempio, quando l'input di abilitazione %I0001 è ON, viene richiamata la funzione SVCREQ No. 7, con l'inizio blocco parametro su %R0001. La bobina output %Q0001 passa a ON se l'operazione ha successo.

SVCREQ 1: Modifica/Leggi il Timer Sweep Costante

Usare SVCREQ 1 per abilitare o disabilitare il modo Sweep a Tempo Costante, modificare la durata dello Sweep a Tempo Costante, Leggere il tempo di Sweep Costante, vedere se lo Sweep a Tempo Costante è al momento abilitato e leggere da durata dello Sweep a Tempo Costante.

Input Blocco Parametro per SVCREQ 1

Per questa funzione, il blocco parametro è lungo due word.

Disabilitazione del Modo Sweep a Tempo Costante

Per disabilitare il modo Sweep a Tempo Costante, immettere la funzione SVCREQ #1 con il seguente blocco parametro:

indirizzo	0
indirizzo + 1	sconosciuto

Abilitazione del Modo Sweep a Tempo Costante

Per abilitare il modo Sweep a Tempo Costante, immettere la funzione SVCREQ #1 con il seguente blocco parametro:

indirizzo	1
indirizzo + 1	0 o valore del timer

Note: Se il timer dovesse usare un nuovo valore, immetterlo nella seconda word. Se il timer rimane invariato, nella seconda word immettere 0. Se il valore del timer non esiste ancora, immettendo 0 la funzione imposterà l'output OK su OFF.

Modifica del Tempo di Sweep a Tempo Costante

Per modificare il valore timer **senza** modificare la selezione delle condizioni del modo sweep, immettere la funzione SVCREQ #1 con il blocco parametro:

indirizzo	2
indirizzo + 1	nuovo valore timer

Lettura delle Condizioni e della Durata dello Sweep a Tempo Costante

Per leggere la durata e le condizioni correnti senza alcuna modifica, immettere la funzione SVCREQ #1 con il seguente blocco parametro:

indirizzo	3
indirizzo + 1	sconosciuto

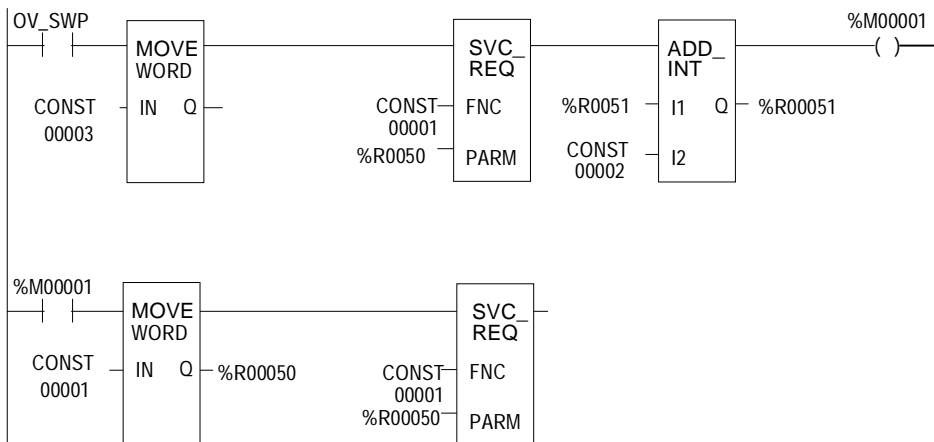
L’esecuzione avrà successo, a meno che:

1. Per l’operazione richiesta venga immesso un No. diverso da 0, 1, 2 o 3:
2. Il tempo di sweep sia superiore a 500ms (0.5 secondi).
3. Il tempo di sweep costante sia abilitato senza il tempo timer programmato o con un vecchio valore 0 per il timer.

Dopo l’esecuzione della funzione, questa riporta le condizioni de timer e il valore agli stessi riferimenti del blocco parametro:

	0 = disabilitato
indirizzo	1 = abilitato
indirizzo + 1	valore timer corrente

Esempio di SVCREQ 1

In questo esempio, se il contatto OV_SWP è impostato, viene letto il Timer Sweep Costante, il timer viene incrementato di due ms. e il nuovo tempo timer viene inviato al PLC. Il blocco parametro è nella memoria locale nella locazione %R0050. Dato che le funzioni MOVE e ADD richiedono tre locazioni di contatto orizzontali, la logica dell’esempio usa la bobina interna discreta %M00001 come locazione temporanea perché il risultato della linea del primo gradino abbia successo. In qualsiasi sweep con OV_SWP non impostato, %M00001 passa a off.

SVCREQ 2: Lettura della Finestra Tempi

SVCREQ 2 può essere utilizzato per leggere i tempi della finestra comunicazioni col programmatore e di quella di comunicazioni col sistema. Queste finestre possono operare nel modo Limitato o nel modo Fino alla Fine (Run to Completion).

Nome Modo	Valore	Descrizione
Modo Limitato	0	Il tempo di esecuzione della finestra è limitato a 6ms. La finestra termina quando non ha più compiti da svolgere o allo scadere dei 6ms.
Modo Run to Completion	2	Indipendentemente dal tempo assegnato a una finestra, procede fino al temine di tutti i compiti della finestra (fino a 400ms).

Una finestra viene disabilitata quando il valore del tempo è zero.

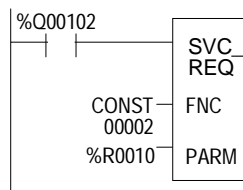
Output del Blocco parametro per SVCREQ 2

Il blocco parametro è lungo tre word:

	Byte Alto	Byte Basso	
indirizzo	Modo	Valore in ms	Finestra Programmatore
indirizzo + 1	Modo	Valore in ms	Finestra Comunicazioni Sistema
indirizzo + 2	deve essere 0	deve essere 0	riservato

Tutti i parametri sono parametri output. Per programmare questa funzione non è necessario immettere alcun valore nel blocco parametro.

Esempio di SVCREQ 2

Nell’esempio che segue, quando viene abilitato l’output %Q00102, la CPU pone i valori di tempo correnti delle finestre nel blocco parametro, partendo dalla locazione %R0010.

SVCREQ 3: Modifica del Modo Finestra Comunicazioni col Programmatore

Usare SVCREQ 3 per cambiare il modo finestra comunicazioni col programmatore (Limitato o Fino alla Fine). Il cambiamento avviene durante lo sweep successivo della CPU, dopo il richiamo della funzione. Il tempo della finestra non può essere cambiato: rimane sempre 6ms.

SVCREQ 3 passa il flusso di corrente a destra, a meno che venga selezionato un modo che non sia 0 (Limitato) o 2 (Fino alla Fine).

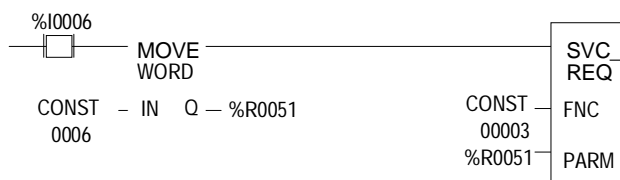
Il blocco parametro è lungo una word.

Modifica del Modo Finestra Comunicazioni Programmatore

Per modificare la finestra programmatore, immettere SVCREQ 3 con il seguente blocco parametro:

	Byte Alto	Byte Basso
indirizzo	Modo	6

Esempio di SVCREQ 3

Nell'esempio che segue, quando l'impulso di abilitazione %I006 passa a ON, la finestra comunicazioni programma viene abilitata e le viene assegnato un valore di 6ms. Il blocco parametro è nella memoria di riferimento alla locazione %R0051.

SVCREQ 4: Modifica del Modo Finestra Comunicazioni col Sistema

Usare SVCREQ 4 per cambiare il modo finestra comunicazioni col sistema (Limitato o Fino alla Fine). Il cambiamento avviene durante lo sweep successivo della CPU, dopo il richiamo della funzione. Il tempo della finestra non può essere cambiato: rimane sempre 6ms.

SVCREQ 4 passa il flusso di corrente a destra, a meno che venga selezionato un modo che non sia 0 (Limitato) o 2 (Fino alla Fine).

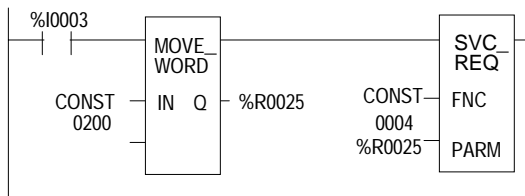
Il blocco parametro è lungo una word.

Modifica del Modo Finestra Comunicazioni Sistema

Per modificare la finestra programmatore, immettere SVCREQ 4 con questo blocco parametro:

	Byte Alto	Byte Basso
indirizzo	Modo	6

Esempio di SVCREQ 4

Nell'esempio che segue, quando l'impulso di abilitazione %I0003 passa a ON, la finestra comunicazioni sistema passa al nodo Fino alla Fine. Il blocco parametro è nella locazione %R0025.

SVCREQ 6: Cambia/Leggi il Numero di Word per la Checksum

Usare SVCREQ 6 per leggere o cambiare il numero di word del programma per la checksum. La funzione avrà successo, a meno che come operazione richiesta venga immesso un numero diverso da 0 o 1.

Formati Blocco Parametro per SVCREQ 6

Il blocco parametro è lungo due word.

Per leggere il conteggio word, la prima word del blocco parametro deve contenere uno zero:

indirizzo	0 (lettura conteggio word)
indirizzo + 1	ignorato

La funzione restituisce il conteggio word corrente nella seconda word del blocco parametro.

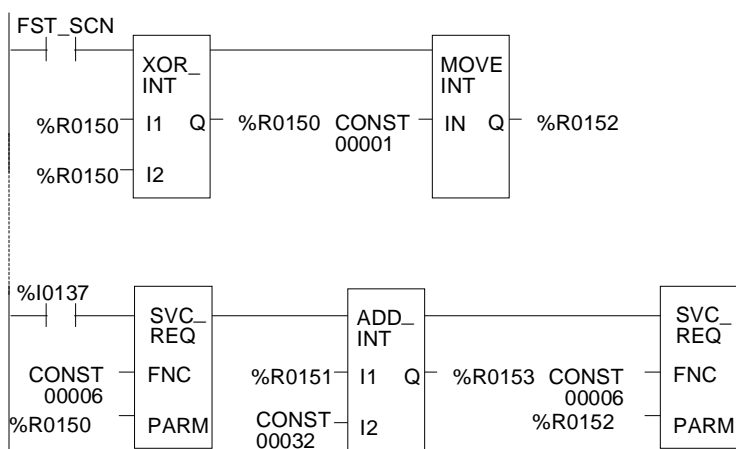
indirizzo	0
indirizzo + 1	conteggio word corrente

Per cambiare il conteggio word, la prima word del parametro deve contenere un 1:

indirizzo	1 (cambio conteggio word)
indirizzo + 1	nuovo conteggio word (0 o 32)

Il PLC cambierà il numero di word per la checksum con un nuovo valore.

Esempio di SVCREQ 6

Nell'esempio che segue, quando viene abilitato il contatto FST_SCN, vengono costruiti i blocchi parametro per la funzione checksum. Successivamente, nel programma, se l'input %I0137 viene attivato, SVCREQ legge il numero di word per la checksum. Il blocco parametro per la funzione Leggi è allocato a %R0150-151. La funzione ADD aggiunge 32 al conteggio word corrente in %R0151 e pone il risultato in %R0153. Il blocco parametro per la funzione Cambia è allocato a %R00152-153. Quindi il secondo SVCREQ inserisce il nuovo conteggio word specificato in %R0153.

SVCREQ 7: Lettura o Modifica di Data e Ora

Usare SVCREQ 7 per leggere o cambiare la data e l'ora nell'orologio del PLC. I dati possono essere sia in BCD che in ASCII. L'anno può essere nel formato a 2 cifre o a 4 cifre. La funzione avrà successo a meno che, per l'operazione richiesta, venga immesso un numero diverso da 0 (leggi) o 1 (cambia), oppure se viene specificato un formato dati non valido o anche se i dati vengono forniti in un formato non previsto.

Formato Blocco Parametro per SVCREQ 7

Per le funzioni data/ora, la lunghezza del blocco parametro dipende dal formato dei dati. Il blocco dati può essere in BCD o in ASCII. Il formato BCD richiede 6 word, il formato ASCII packed richiede 12 word (13 word per l'anno a 4 cifre). Per entrambi i tipi di dati:

- Le ore vengono caricate nel formato a 24 ore.
- Il giorno della settimana è in valore numerico da 1 (domenica) a 7 (sabato)

	Formato Anno a 2 Cifre	**Formato Anno a 4 Cifre**
indirizzo	0 = leggi ora e data 1 = imposta ora e data	0 = leggi ora e data 1 = imposta ora e data
indirizzo + 1	1 = Formato BCD 3 = Formato ASCII packed	81h = Formato BCD 83h = Formato ASCII packed
indirizzo da + 2 alla fine	dati	dati

Le word dalla 3 alla fine del blocco parametro contengono i dati output di ritorno dalla funzione di lettura o i nuovi dati forniti dalla funzione cambia. In entrambi i casi, il formato di queste word è il medesimo. Quando si leggono la data e l'ora, le word (indirizzo +2) sull'input alla fine del blocco parametro vengono ignorate.

SVCREQ 7 Contenuto del Blocco Parametro: Formato BCD

Nel formato BCD, ciascuna voce ora e data occupa un byte, pertanto il blocco parametro è di sei word.

Anno a 2 cifre

L'ultimo byte della sesta word non viene usato. Quando si impostano la data e l'ora, questo byte viene ignorato; quando si leggono la data e l'ora, la funzione ritorna a 00.

Formato Blocco Parametro:
Byte Alto: Byte Basso

Esempio:
Leggi Data e Ora nel formato BCD
(Dom., 3 Lugl, 1998, ore 2:45:30 p.m.)

Byte Alto	Byte Basso		Esempio (alto)	Esempio (basso)
1 = cambia o 0 = leggi		indirizzo	0 (leggi)	
1 (Formato BCD)		indirizzo + 1	1 (formato BCD)	
mese	anno	indirizzo + 2	07 (luglio)	98 (anno)
ore	giorno del mese	indirizzo + 3	14 (ore)	03 (giorno)
secondi	minuti	indirizzo + 4	30 (secondi)	45 (minuti)
(nullo)	giorno della settimana	indirizzo + 5	00	06 (venerdì)

Anno a 4 Cifre

Il blocco parametri è di sei word. Vengono usati tutti i byte.

Formato Blocco Parametro:
Byte Alto: Byte Basso

Esempio:
Leggi Data e Ora in formato BCD
(Dom., 3 Lugl, 1998, ore 2:45:30 p.m.)

Byte Alto	Byte Basso		Esempio (alto)	Esempio (basso)
1 = cambia o 0 = leggi		indirizzo	00	00 (leggi)
81h (Formato BCD, 4 cifre)		indirizzo + 1	00	81h (BCD, 4-digit)
anno	anno	indirizzo + 2	19 (anno)	98 (anno)
giorno del mese	mese	indirizzo + 3	03 (giorno)	07 (luglio)
minuti	ore	indirizzo + 4	45 (minuti)	14 (ore)
giorno della settimana	secondi	indirizzo + 5	06 (venerdì)	30 (secondi)

SVCREQ 7 Contenuto Blocco Parametro: Formato ASCII Packed

Nel formato ASCII packed, ciascuna cifra dell'ora e della data è in byte in formato ASCII. Gli spazi e le colonne, per stampare e visualizzare, sono inclusi nei dati. Il formato ASCII richiede 13 word nel blocco parametri (13 per l'anno a 4 cifre).

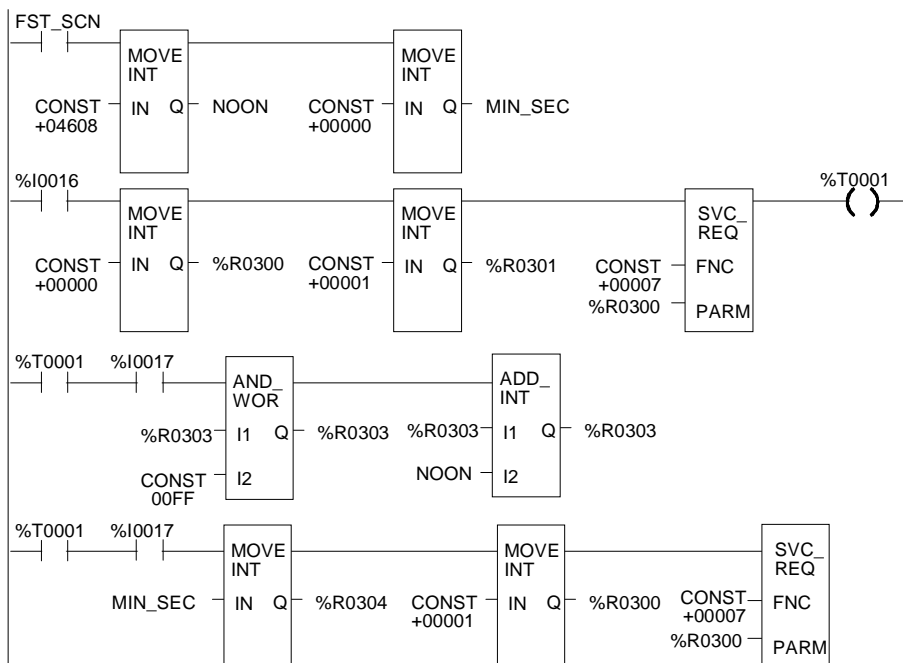
Anno a 2 Cifre

Formato Blocco Parametro: Byte Alto	Byte Basso		Esempio: Leggi Data e Ora nel Formato ASCII Packed (Lun, 5 Ott, 1998 ore 11:13:00pm)	
1 = cambia or 0 = leggi		indirizzo	0 (leggi)	
3 (ASCII format)		indirizzo + 1	3 (formato ASCII)	
anno	anno	indirizzo + 2	38 (8)	39 (9)
mese	(spazio)	indirizzo + 3	31 (1)	20 (spazio)
(spazio)	mese	indirizzo + 4	20 (spazio)	30 (0)
giorno del mese	giorno del mese	indirizzo + 5	35 (5)	30 (0 iniziale)
ore	(spazio)	indirizzo + 6	31 (1)	20 (spazio)
:	ore	indirizzo + 7	3A (:)	31 (1)
minuti	minuti	indirizzo + 8	33 (3)	31 (1)
secondi	:	indirizzo + 9	30 (0)	3A (:)
(spazio)	secondi	indirizzo + 10	20 (space)	30 (0)
giorno della sett.	giorno della set.	indirizzo + 11	32 (2: Lun.)	30 (0 iniziale)

Anno a 4 Cifre

Formato Blocco Parametro: Byte Alto	Byte Basso		Esempio: Leggi Data e Ora nel Formato ASCII Packed (Lun, 5 Ott, 1998 ore 11:13:00pm)	
1 = cambia o 0 = leggi		indirizzo	0 (leggi)	
83h (ASCII 4 cifre)		indirizzo + 1	83h (ASCII 4 cifre)	
anno (centinaia)	anno (migliaia)	indirizzo + 2	39 (9)	31 (1)
anno (unità)	anno (decine)	indirizzo + 3	38 (8)	39 (9)
mese (decine)	(spazio)	indirizzo + 4	31 (1)	20 (spazio)
(spazio)	mese (unità)	indirizzo + 5	20 (spazio)	30 (0)
giorno del mese (unità)	giorno del mese (decine)	indirizzo + 6	35 (5)	30 (0 iniziale)
ore (decine)	(spazio)	indirizzo + 7	31 (1)	20 (spazio)
: (colonna)	ore (unità)	indirizzo + 8	3A (:)	31 (1)
minuti (unità)	minuti (decine)	indirizzo + 9	33 (3)	31 (1)
secondi (decine)	: (colonna)	indirizzo + 10	30 (0)	3A (:)
(spazio)	secondi (unità)	indirizzo + 11	20 (space)	30 (0)
giorno della settimana (unità)	giorno della settimana (decine)	indirizzo + 12	32 (2: Lun.)	30 (0 iniziale)

Esempio di SVCREQ 7

Nell'esempio che segue, quando richiamato da una logica precedente, viene costruito un blocco parametro per il giorno e l'ora, che richiede la data e l'ora correnti, quindi imposta l'orologio a mezzogiorno in formato BCD. Il blocco parametro viene allocato nella locazione %R0300. L'array NOON (mezzogiorno) è stato impostato altrove nel programma, perché contenga i valori 12, 0 e 0. (l'Array NOON deve anche contenere i dati in %R0300). Il formato BCD richiede sei locazioni di memoria contigue per il blocco parametro.

SVCREQ 8: Reset del Timer Watchdog

Usare SVCREQ 8 per resettare il timer watchdog durante lo sweep. Normalmente, quando scade il tempo del timer watchdog, il PLC si spegne senza avvertimenti. SVCREQ 8 consente al timer di continuare a funzionare durante un compito che richiede tempo (per esempio, in attesa di una risposta da una linea di comunicazione).

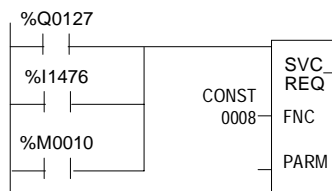
Attenzione

Verificare che il resetaggio del timer watchdog non influenzi negativamente il processo di controllo.

Formato Blocco Parametro per SVCREQ 8

A questa funzione non è associato un blocco parametro.

Esempio di SVCREQ 8

Nell'esempio che segue, la corrente passa attraverso l'input di abilitazione %Q0027 o l'input %I1476 o la bobina interna %M00010, causando il reset del timer watchdog.

SVCREQ 9: Lettura Tempo Sweep dall'inizio dello Sweep

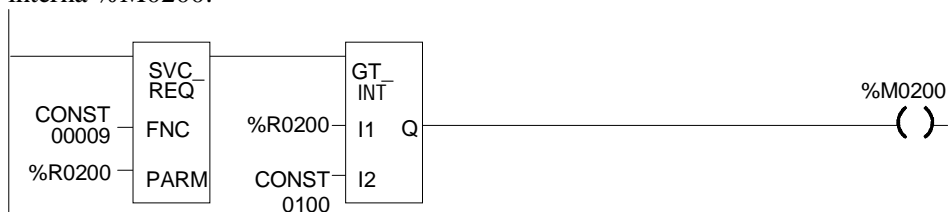
Usare SVCREQ 9 per leggere il tempo in millisecondi dall'inizio dello sweep. Il formato dati è un integrale a 16 bit senza segno.

Formato parametro Output per SVCREQ 9

Il blocco parametro è solo un blocco parametro output, lungo una word.

indirizzo	tempo dall'inizio dello sweep

Esempio di SVCREQ 9

Nell'esempio che segue, il tempo trascorso dall'inizio dello sweep viene sempre letto nella locazione %R0200. Se è superiore a 100 ms, viene attivata la bobina interna %M0200.

SVCREQ 10: Lettura Nome Folder

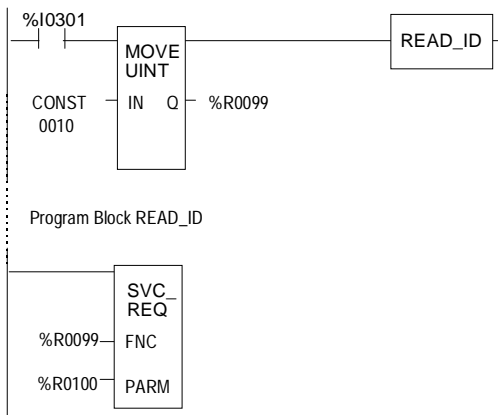
Usare SVCREQ 10 per leggere il nome del folder al momento in esecuzione.

Formato Blocco parametro Output per SVCREQ 10

Il blocco parametro output è lungo quattro word e restituisce otto caratteri ASCII, dove l'ultimo è un carattere nullo (00h). Se il nome del programma è formato da meno di sette caratteri, alla fine vengono aggiunti caratteri nulli

	Byte Basso	Byte Alto
indirizzo	carattere 1	carattere 2
indirizzo + 1	carattere 3	carattere 4
indirizzo + 2	carattere 5	carattere 6
indirizzo + 3	carattere 7	00

Esempio di SVCREQ 10

Nell'esempio che segue, quando l'input di abilitazione %I0301 passa a OFF, nella locazione del registro %R0099 viene caricato il valore 10, che è il codice per la funzione Leggi Nome Folder. Viene così richiamato il Blocco Programma READ_ID per recuperare il nome del folder. Il blocco parametro viene allocato all'indirizzo %R0100.

SVCREQ 11: Leggi l’ID PLC

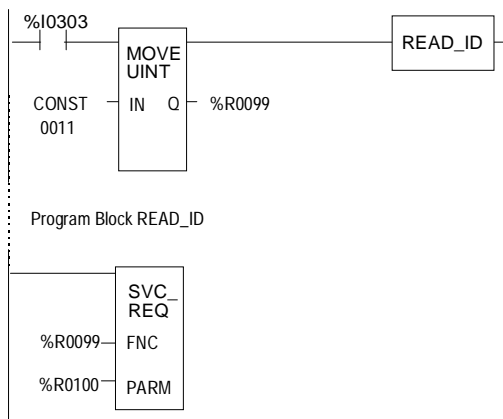
Usare SVCREQ 11 per leggere il nome del PLC che esegue il programma.

Formato Blocco Parametro Output per SVCREQ 11

Il blocco parametro output è lungo quattro word e restituisce otto caratteri ASCII, dove l’ultimo è un carattere nullo (00h). Se l’ID PLC è formato da meno di sette caratteri, alla fine vengono aggiunti caratteri nulli

	Byte Basso	Byte Alto
indirizzo	carattere 1	carattere 2
indirizzo + 1	carattere 3	carattere 4
indirizzo + 2	carattere 5	carattere 6
indirizzo + 3	carattere 7	00

Esempio di SVCREQ 11

Nell’esempio che segue, quando l’input di abilitazione %I0303 passa a OFF, nella locazione del registro %R0099 viene caricato il valore 11, che è il codice per la funzione Leggi ID PLC. Viene così richiamato il Blocco Programma READ_ID per recuperare l’ID. Il blocco parametro viene allocato all’indirizzo %R0100.

SVCREQ 13: Spegni (Arresta) il PLC

Usare SVCREQ 13 per arrestare il PLC *al termine dello sweep successivo*. All'inizio dello sweep successivo, tutti gli output passano allo stato designato per default. Nella tabella Errori PLC viene posta un'informazione di errore "Spegni PLC". La scansione di I/O continua come configurata.

Blocco parametro per SVCREQ 13

Questa funzione non dispone del blocco parametro.

Esempio di SVCREQ 13

Nell'esempio che segue, avviene una "Perdita del Modulo di I/O" e viene eseguito SVCREQ 13. L'input PARM non viene utilizzato

Questo esempio usa un JUMP alla fine del programma per forzare lo spegnimento se la funzione Spegni PLC viene eseguita con successo. Questi JUMP e LABEL sono necessari perché la transizione al modo Stop non avviene fino al termine dello sweep nel quale viene eseguita la funzione.

LOS_MD
%T0001
%T0001
SVC_
REQ
>> END_PRG
CONST
00013
FNC
PARM
END_PRG

SVCREQ 14: Cancella Errori

Usare SVCREQ 14 per cancellare la tabella errori PLC o la tabella errori di I/O. L'output SVCREQ viene impostato su ON, a meno che per l'operazione richiesta venga immesso un numero che non sia 0 o 1.

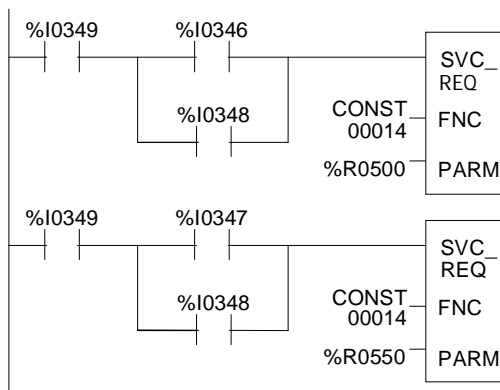
Blocco Parametro Input per SVCREQ 14

Per questa funzione, il blocco parametro è lungo una word. È soltanto un blocco parametro input. Non vi è alcun parametro output.

0 = cancella la tabella errori PLC.
1 = cancella la tabella errori di I/O.

Esempio di SVCREQ 14

Nell'esempio che segue, quando l'input %I0346 è on e l'input %I0349 è on, viene cancellata la tabella errori PLC. Quando l'input %I0347 è on e l'input %I0349 è on, viene cancellata la tabella errori di I/O. Quando l'input %I0348 è on e l'input %I0349 è on, vengono cancellate entrambe le tabelle.

Il blocco parametro per la tabella errori PLC è allocato a %R0500; per la tabella errori di I/O il blocco parametro è allocato a %R0550. Entrambi i blocchi parametro vengono impostati altrove nel programma.

SVCREQ 15: Lettura Ultimo Errore Immesso nella Tabella

Usare SVCREQ 15 per leggere l'ultima immissione nella tabella errori PLC o I/O. L'output SVCREQ è impostato su ON a meno che per l'operazione richiesta venga immesso un numero che non sia 0 o 1 o se la tabella errori è vuota.

Blocco Parametro Input per SVCREQ 15

Per questa funzione, il blocco parametro è lungo 22 word. Il blocco parametro input ha il seguente formato:

	Formato Anno a 2 Cifre	Formato Anno a 4 Cifre
indirizzo	0 = Leggi tabella errori PLC.	8 = Leggi tabella errori PLC.
	1 = Leggi tabella errori I/O.	9 = Leggi tabella errori I/O.

Il formato del blocco parametro output dipende da dove la funzione legge i dati: dalla tabella errori PLC o dalla tabella errori I/O.

Formato Output Tabella Errori PLC: Byte Alto	Byte Basso		Formato Output Tabella Errori I/O: Byte Alto	Byte Basso
0			1	
di scorta	lungo/corto	indirizzo + 1	tipo di memoria	lungo/corto
di scorta	di scorta	indirizzo + 2		offset
slot	rack	indirizzo + 3	slot	rack
	compito	indirizzo + 4	blocco	bus
azione errore	gruppo errore	indirizzo + 5		punto
	codice errore	indirizzo + 6	azione errore	gruppo errore
	dati specifici errore	indirizzo + 7	tipo di errore	categoria di errore
		indirizzo + 8 to indirizzo + 18	dati specifici errore	descrizione errore
minuti	secondi	indirizzo + 19	minuti	secondi
giorno del mese	ora	indirizzo + 20	giorno del mese	ora

Formato Anno a 2 Cifre

Byte Alto	Byte Basso		Byte Alto	Byte Basso
anno	mese	indirizzo + 21	anno	mese

oppure

Formato Anno a 4 Cifre

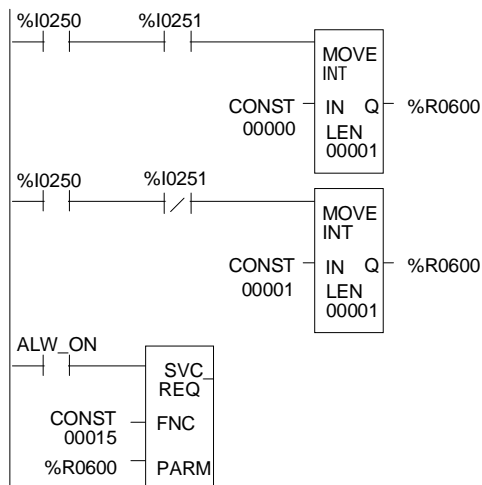
Byte Alto	Byte Basso		Byte Alto	Byte Basso
di scorta	mese	indirizzo + 21	di scorta	mese
anno		indirizzo + 22	anno	

Valore Lungo/Corto

Il primo byte della word indirizzi +1 contiene un numero che indica la lunghezza dei dati specifici dell'errore nell'immissione errore. I valori possibili sono:

Tabella errori PLC.	00 = 8 byte (corto) 01 = 24 byte (lungo)
Tabella errori I/O.	02 = 5 byte (corto) 03 = 21 byte (lungo)

Esempio di SVCREQ 15

Quando gli input %I0250 e %I0251 sono entrambi on, la prima funzione Move pone uno zero (legge la tabella errori PLC) nel blocco parametro per SVCREQ 15. Quando l'input %I0250 è on e l'input %I0251 è off, l'istruzione Move, pone invece un uno (legge la tabella errori di I/O) nel blocco parametro SVCREQ. Il blocco parametro è allocato alla locazione %R0600.

SVCREQ 16 Lettura Tempo Trascorso

Usare SVCREQ 16 per leggere il tempo trascorso dall'orologio del sistema, che misura in secondi il tempo trascorso dall'accensione del PLC.

Blocco parametro Output per SVCREQ 16

Questa funzione ha solo un blocco parametro output, lungo 3 word.

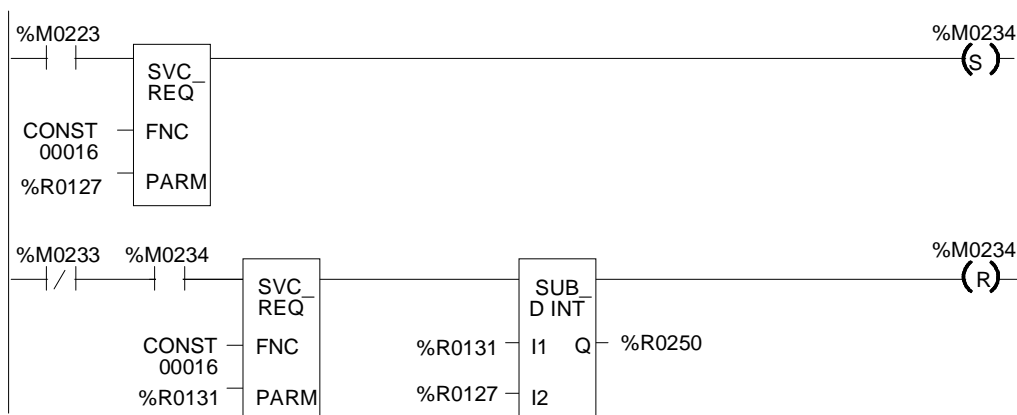
indirizzo	secondi dall'accensione (ordine inf.)
indirizzo + 1	secondi dall'accensione (ordine sup.)
indirizzo + 2	scatti da 100 microsecondi

Le prime due word sono il tempo trascorso in secondi. L'ultima word è il numero di scatti da 100 microsecondi del secondo corrente.

Esempio di SVCREQ 16

Nell'esempio che segue, quando la bobina interna %M0233 è on, il SVCREQ con un blocco parametro allocato a %R0127 legge il tempo trascorso dall'orologio del sistema e imposta la bobina interna %M0234. quando la bobina %M0233 è off, il SVCREQ con un blocco parametro a %R0131 rilegge in tempo trascorso.

La funzione di sottrazione trova la differenza tra la prima e la seconda lettura, che è stata caricata nei blocchi parametro SVCREQ. La sottrazione ignora gli scatti da cento microsecondi.

La differenza tra le due lettura viene posta nella locazione di memoria %R0250.

SVCREQ 18: Lettura Stato di Override I/O

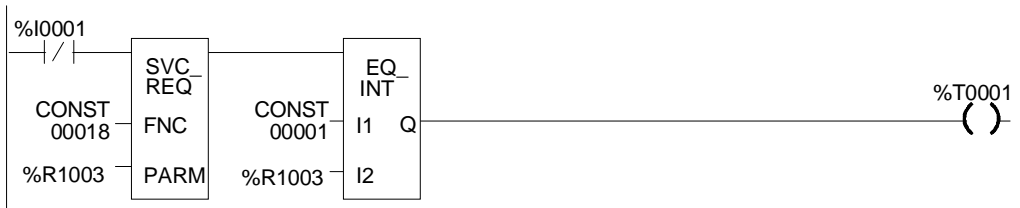
Usare SVCREQ 18 per controllare qualsiasi override nelle memorie della CPU %I e %Q.

Blocco Parametro Output per SVCREQ 18

Questa funzione ha solo un blocco parametro output, lungo una word.

indirizzo	0 = Nessun override impostato. 1 = Override impostati.

Esempio di SVCREQ 18

Il seguente SVCREQ legge lo stato degli override della memoria di I/O nella locazione %R1003. La funzione di uguaglianza controlla %R1003 per vedere se è uguale (alla costante) 1. Se lo è, la funzione uguaglianza attiva un output %T0001.

SVCREQ 23: Lettura della Master Checksum

Usare SVCREQ 23 per leggere la master checksum del programma applicativo e della configurazione. Se la funzione è abilitata, l'output SVCREQ è sempre ON.

Blocco Parametro Output per SVCREQ 23

Per questa funzione, il blocco parametro output è lungo 12 word con il seguente formato:

Le prime due voci nel blocco parametro output indicano quando le checksum del programma e della configurazione sono valide. (Le checksum del programma potrebbero non essere valida durante un Run Mode Store.)

indirizzo	Master Checksum Programma Valida (0 = non valida, 1 = valida)
indirizzo + 1	Master Checksum Configuraz. Valida (0 = non valida, 1 = valida)
indirizzo + 2	Numero di Blocchi Programma (compreso _MAIN)
indirizzo + 3	Dimensioni del Programma utente in Byte (dati tipo DWORD)
indirizzo + 5	Checksum Programma Aggiuntivo
indirizzo + 6	Checksum Programma CRC (dati tipo DWORD)
indirizzo + 8	Dimensioni dei Dati di Configurazione in Byte
indirizzo + 9	Checksum Configurazione Aggiuntiva
indirizzo + 10	Checksum Configurazione CRC (dati tipo DWORD)

Esempio di SVCREQ 23

Nell'esempio che segue, quando l'input %I0251 è ON, l'informazione della master checksum viene posta nel blocco parametro a %R0050 e la bobina output (%Q0001) viene attivata.

SVCREQ 26/30: Interrogazione I/O

Usare i SVCREQ 26 e 30 per controllare se i moduli installati rispondono alla configurazione software. In caso contrario, questo SVCREQ pone gli appropriati errori di addizione, sottrazione e non rispondenza nelle tabelle errori PLC o I/O. I SVCREQ 26 e 30 eseguono entrambi le medesime funzioni.

Più tanti sono gli errori di configurazione, maggiore sarà il tempo di esecuzione di questi SVCREQ.

Questi SVCREQ non hanno blocco parametro e inviano sempre flusso di corrente.

Esempio di SVCREQ 26

Nell'esempio che segue, quando l'input %I0251 è ON, SVCREQ controlla i moduli installati e li confronta con la configurazione software. Terminato il SVCREQ, viene attivato l'output %Q0001.

SVCREQ 29: Lettura Tempo PLC Spento

Usare SVCREQ 29 per leggere il tempo trascorso tra l'ultimo spegnimento e la più recente accensione. Se il tempo del timer watchdog è scaduto prima dello spegnimento, il PLC non sarà in grado di leggere il tempo di spegnimento, pertanto il tempo viene azzerato.

L'output SVCREQ è sempre ON.

Blocco parametro Output per SVCREQ 29

Questa funzione ha solo un blocco parametro output, lungo 3 word.

indirizzo	Tempo di Spegnimento in Secondi (ordine inf.)
indirizzo + 1	Tempo di Spegnimento in Secondi (ordine sup.)
indirizzo + 2	zero

Le prime due word sono il tempo di spegnimento in secondi. L'ultima word è sempre 0.

Esempio di SVCREQ 29

Nell'esempio che segue, quando l'input %I0251 è ON, il tempo di spegnimento viene posto nel blocco parametro che inizia a %R0050. Viene attivata la bobina output (%Q0001).

Capitolo 9

Protocolli I/O/SNP/RTU Seriali

Questo capitolo descrive le caratteristiche del I/O seriale della CPU, che può essere utilizzato per leggere/scrivere le attività per una delle porte CPU, direttamente dal programma applicativo.

Questo capitolo contiene anche le istruzioni per utilizzare i COMMREQ per configurare le porte seriali CPU per il protocollo SNP, RTU, o di I/O Seriale.

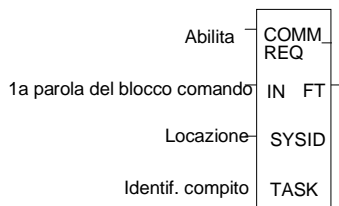
- Formato della funzione COMMREQ
- Configurazione delle Porte Seriali con la Funzione COMMREQ
  - RTU Slave/SNP Operazione Slave con un Programmatore Connesso
  - Blocco Comando COMMREQ per Configurare il Protocollo SNP
  - Blocco Dati COMMREQ per Configurare il Protocollo RTU
  - Blocco Dati COMMREQ per Configurare l'I/O Seriale
- Comandi COMMREQ per I/O Seriale
  - Inizializza Porta
  - Imposta il Buffer Input
  - Flusso Buffer Input
  - Lettura Stato Porta
  - Scrittura Controllo Porta
  - Annulla Operazione
  - Composizione Automatica
  - Scrittura Byte
  - Lettura Byte
  - Lettura Stringa

I dettagli dei protocolli RTU e SNP sono descritti nel *manuale Utente Comunicazioni Seriali* (GFK-0582).

Formato della Funzione Richiesta di Comunicazione

L’I/O seriale viene implementato dalle funzioni Richiesta di Comunicazione (COMMREQ). Le operazioni del protocollo, come la trasmissione di un carattere tramite la porta seriale o l’attesa dell’input di un carattere, vengono implementate mediante il blocco funzione COMMREQ.

COMMREQ richiede che tutti i suoi dati comandi siano posti nell’ordine corretto (in un *blocco comando*) nella memoria CPU prima che venga eseguito. COMMREQ dovrebbe essere poi eseguito dal contatto di una bobina “one-shot” per evitare che venga inviato varie volte. Per spostare le parole e creare un blocco comando nelle tabelle di Registro, si dovrebbe usare una serie di comandi Muovi Blocco (BLKMV).

La funzione COMMREQ ha tre input e un output. Quando la funzione riceve flusso di corrente, un blocco comando di dati viene inviato al modulo specificato.

Parametri della Funzione COMMREQ

Input/ Output	Scelte	Descrizione
abilita	flusso	Quando la funzione riceve corrente, viene eseguita la richiesta di comunicazione.
IN	R, AI, AQ	IN contiene la prima word del blocco comando.
SYSID	I, Q, M, T, G, R, AI, AQ, costanti	SYSID contiene il numero di rack (il byte più significativo) e il numero di slot (l’ultimo byte significativo) del dispositivo bersaglio. Per CPU, SYSID di deve specificare rack/slot 0.
TASK	R AI, AQ, costanti	TASK specifica la porta cui è destinata l’operazione: compito 19 per la porta 1 compito 20 per la porta 2
FT	flusso, no	FT viene attivato se in COMMREQ viene rilevato un errore: • Non c’è l’indirizzo del bersaglio specificato (SYSID). • Il compito specificato non è valido per il dispositivo (TASK). • La lunghezza dei dati è 0. • Non c’è l’indirizzo (nel blocco comando) del puntatore dello stato del dispositivo.

Blocco Comando per la Funzione COMMREQ

Il Blocco Comando inizia al riferimento specificato nel parametro IN di COMMREQ. La lunghezza del Blocco Comando dipende dalla quantità di dati inviati al dispositivo.

Il Blocco Comando contiene i dati da comunicare all'altro dispositivo, oltre alle informazioni riguardanti l'esecuzione di COMMREQ. IL Blocco Comando ha la seguente struttura:

indirizzo	Lunghezza (in word)
indirizzo + 1	Flag Attesa/Non Attesa
indirizzo + 2	Memoria Puntatore Stato
indirizzo + 3	Offset Puntatore Status
indirizzo + 4	Valore Timeout di Riposo
indirizzo + 5	Tempo massimo di Comunicazione
indirizzo da + 6 a indirizzo + 133	Blocco Dati

Esempio della Funzione COMMREQ

Nell'esempio che segue, quando %M0021 è ON, un Blocco Comando allocato a %R0032 viene inviato alla porta 2 (compito di comunicazione 20) della CPU (rack 0, slot 0). Se nell'elaborazione di COMMREQ avviene un errore, viene impostato %Q0110.

%M0021
COMM_REQ
%R0032 IN FT
%Q0110
CONST 0000 SYSID
CONST 00020 TASK

Configurazione Porte Seriali con la Funzione COMMREQ

Le tabelle che seguono elencano i valori del Blocco Comando richiesti per impostare una Porta Seriale per SNP, RTU, e l'I/O seriali. Tutti i valori sono in esadecimali, se non diversamente indicato. I comandi BLKMV che vengono usati per creare un Blocco Comando sono descritti in questo esempio.

Temporizzazione

Se una configurazione porta COMMREQ viene inviata a una porta cui al momento è collegato un SNP/SNPX master (per esempio, il programmatore), la configurazione della porta seriale specificata da COMMREQ non avrà luogo fino a che la CPU rileva lo scollegamento dal SNP/SNPX master. Ciò avviene al tempo T3' dopo lo scollegamento del master. La word di stato COMMREQ per la configurazione porta COMMREQ viene aggiornata non appena la CPU verifica che la configurazione specificata è valida. Ciò significa che un valore di Successo COMMREQ può essere reinviato alla Configurazione Porta COMMREQ prima che la configurazione specificata sia effettivamente installata.

Invio di un altro COMMREQ alla Stessa Porta

Il programma applicativo deve attendere almeno 2 secondi dopo l'installazione del protocollo di una nuova porta seriale prima di inviare alla porta un COMMREQ specifico per quel protocollo. Ciò vale per un nuovo protocollo installato caricando una nuova configurazione hardware o con una configurazione COMMREQ della porta. Se la porta viene configurata per l'I/O seriale, questo periodo di attesa deve far seguito anche a qualsiasi transizione del modo Stop to Run della CPU.

Combinazioni non Valide per la Configurazione della Porta

Le configurazioni di entrambe le porte devono essere compatibili. La CPU rifiuta le seguenti combinazioni:

Porta 1	*Porta 2*
Disabilitata	Disabilitata
Disabilitata	I/O Seriale (Commut. CPU Run/Stop disabilitato)
I/ O Seriale (Commut. CPU Run/Stop disabilitato)	Disabilitato
I/O Seriale (Commut. CPU Run/Stop disabilitato)	I/O Seriale (Commut. CPU Run/Stop disabilitato)

Funzionamento RTU Slave/SNP Slave con Programmatore Collegato

Un programmatore (computer) può essere collegato alla porta 1 o 2 mentre è attivo il modo slave RTU sulla porta. Per connessioni multi-drop, la CPU deve essere configurata per utilizzare un appropriato ID PLC. Si noti che per una connessione SNP multi-drop con la porta al momento configurata per RTU, la impostazione dell'ID SNP associato alla CPU deve corrispondere all'ID multi-drop.

Il programmatore, perché possa essere riconosciuto, deve utilizzare gli stessi parametri di comunicazione seriale (baud-rate, parità, stop bit, ecc.) del protocollo RTU slave correntemente attivo per il programmatore.

Quando la CPU riconosce il programmatore, rimuove il protocollo slave RTU dalla porta e installa il SNP slave come protocollo correntemente attivo. L'ID SNP, il tempo di risposte del modem e il tempo di attesa di default per questa nuova sessione SNP slave vengono rilevati dalle impostazioni della CPU configurata, non quindi per le configurazioni della porta 1 o 2. La connessione deve essere stabilita entro 12 secondi. Una volta stabilita la connessione col programmatore, potrà iniziale la normale comunicazione con il programmatore. (La mancata connessione col programmatore entro 12 secondi viene trattata come una perdita di comunicazione col programmatore).

Il programmatore può inviare un nuovo protocollo COMMREQ mediante configurazione o un COMMREQ Impostazione Porta Seriale. (COMMREQ non supportati da un protocollo SNP slave vengono rifiutati). Se viene ricevuto un nuovo protocollo, non avrà effetto fino a che il programmatore non verrà scollegato.

Dopo lo scollegamento del programmatore, la CPU ne rileverà l'assenza con un certo ritardo (pari al timeout configurato in SNP T3'), durante il quale nessun messaggio viene elaborato sulla porta. La CPU rileva lo scollegamento del programmatore come un timeout del protocollo SNP slave, pertanto è importante prestare attenzione nel disabilitare i timeout usati da un protocollo SNP slave.

Quando la CPU riconosce la connessione, reinstalla il protocollo RTU slave, a meno che non sia stato ricevuto un nuovo protocollo. In tal caso, la CPU installerà il nuovo protocollo.

Esempio

1. La porta 1 sta eseguendo un protocollo RTU slave a 9600 baud.
2. Alla porta 1 viene collegato un programmatore che opera a 9600 baud.
3. La CPU installa un SNP slave sulla porta 1 e il programmatore comunica normalmente.
4. Il programmatore carica una nuova configurazione sulla porta 1,che imposta l'SNP slave a 4800 baud (che non avrà effetto fino a che la porta non perderà la comunicazione col programmatore).
5. Quando la CPU perderà la comunicazione col programmatore, prenderà affetto la nuova configurazione.

Blocco Comando COMMREQ per Configurare il Protocollo SNP

	Valori	Significato
indirizzo	10H	Lunghezza Blocco dati
indirizzo + 1	0 = No Attesa	Flag ATTESA/NO ATTESA
indirizzo + 2	0008 = %R, REGISTRO MEMORIA	Stato word Puntatore Tipo di Memoria
indirizzo + 3	Numero in base zero che dà l'indirizzo della word di stato COMMREQ (per esempio, un malore di 99 dà un indirizzo di 100 per la word di stato)	Status Word Pointer Offset
indirizzo + 4	0 (Usato solo nel modo Attesa/No Attesa)	Valore Timeout di riposo
indirizzo + 5	0 (Usato solo nel modo Attesa/No Attesa)	Tempo Massimo di Comunicazione
indirizzo + 6	FFF0H	parole di Comando (imp. porta seriale)
indirizzo + 7	0001	Protocollo: 1=SNP
indirizzo + 8	0000=Slave	Modo Porta
indirizzo + 9	7=38400, 6=19200, 5=9600, 4=4800	Velocità Dati
indirizzo + 10	0 = None, 1 = Dispari, 2 = pari	Parità
indirizzo + 11	1 = Nessuno	Controllo Flusso
indirizzo + 12	0 = Nessuno, 1 = 10ms, 2 = 100ms, 3 = 500ms	Ritardo tempo di risposta
indirizzo + 13	0 = Long, 1 = Medium, 2 = Short, 3 = None	Timeout
indirizzo + 14	1 = 8 bit	Bit Per Carattere
indirizzo + 15	0 = 1 Stop Bit, 1 = 2 Stop bit	Stop Bit
indirizzo + 16	non in uso	Interfaccia
indirizzo + 17	non in uso	Modo Duplex
indirizzo + 18	fornito dall'utente*	Byte 1 e 2 identificazione dispositivo
indirizzo + 19	fornito dall'utente*	Byte 3 e 4 identificazione dispositivo
indirizzo + 20	fornito dall'utente*	Byte 5 e 6 identificazione dispositivo
indirizzo + 21	fornito dall'utente*	Byte 7 e 8 identificazione dispositivo

* L'identificatore del dispositivo per le porte SNP slave è confezionato con due parole, con l'ultimo carattere significativo nell'ultimo byte significativo della word. Per esempio, se i primi due caratteri sono "A" e "B", l'indirizzo +18 conterrà il valore esadecimale 4241.

Blocco Comando COMMREQ per Configurare il Protocollo RTU

	Valori	Significato
Prime 6 parole		Riservate per l'uso COMMREQ.
indirizzo + 6	FFF0H	Comando
indirizzo + 7	0003	Protocollo: 0003=RTU
indirizzo + 8	0000	Modo Porta: 0000=Seervita
indirizzo + 9	6=19200, 5=9600, 4=4800	Velocità Dati
indirizzo + 10	0 = Nessuno, 1 = Dispari, 2 = Pari	Parità
indirizzo + 11	0 = Hardware, 1 = Nessuno	Controllo Flusso
indirizzo + 12	non in uso	Ritardo tempo di risposta
indirizzo + 13	non in uso	Timeout
indirizzo + 14	non in uso	Bit per Carattere
indirizzo + 15	non in uso	Stop Bit
indirizzo + 16	non in uso	Interfaccia
indirizzo + 17	0 = 2-wire, 1 = 4-wire	Modo Duplex
indirizzo + 18	Indirizzo Stazione (1-247)	Identificatore Dispositivo
indirizzo + 19—21	non in uso	Identificatore Dispositivo

Blocco Comando COMMREQ per Configurare il Protocollo I/O Seriale

	Valori	Significato
Prime 6 Parole		Riservate per l'uso COMMREQ.
indirizzo + 6	FFF0H	Comando
indirizzo + 7	0005	Protocollo: 0005= IO Seriale
indirizzo + 8	0 = Slave	Modo Porta
indirizzo + 9	6=19200, 5=9600, 4=4800	Velocità Dati
indirizzo + 10	0 = Nessuno, 1 = Dispari, 2 = Pari	Parità
indirizzo + 11	0 = Hardware, 1 = Nessuno	Controllo Flusso
indirizzo + 12	0 = Nessuno	Ritardo tempo di risposta
indirizzo + 13	0 = Long	Timeout
indirizzo + 14	0=7 bit, 1=8 bit	Bit per Carattere
indirizzo + 15	0 = 1 stop bit, 1 = 2 stop bit	Stop Bit
indirizzo + 16	non in uso	Interfaccia
indirizzo + 17	0 = 2 fili, 1 = 4 fili	Modo Duplex
indirizzo + 18—21	non in uso	Identificatore Dispositivo

Richiamo dei COMMREQ I/O Seriali dallo Sweep PLC

L’implementazione di un protocollo seriale con i COMMREQ I/O Seriali può essere limitata dal tempo sweep PLC. Per esempio, se il protocollo richiede che la risposta a un certo messaggio dal dispositivo remoto sia iniziata entro 5 mS dal ricevimento del messaggio, questo metodo potrebbe non avere successo se il tempo sweep PLC è 5mS o maggiore, in quanto non è garantita la risposta in tempo utile.

Dato che l’I/O seriale è completamente pilotato dal programma applicativo, nel modo STOP una porta configurata come I/O Seriale si converte immediatamente in SNP slave per facilitare la comunicazione col programmatore. Pertanto, nel modo STOP, il protocollo I/O Seriale non è attivo; è attivo soltanto quando il PLC è nel modo RUN.

Compatibilità

I blocchi di funzione COMMREQ supportati dall’I/O Seriale non sono supportati dagli altri protocolli al momento esistenti (come SNP Slave, SNP Master e RTU Slave). Se si tenta di utilizzarli con una porta configurata per uno di questi protocolli si ha un ritorno di errori.

Word di Stato per i COMMREQ I/O Seriali

Terminato con successo il COMMREQ, nella word di stato COMMREQ viene ritornato il valore di 1. Qualsiasi altro valore reso è un codice di errore, dove il byte basso è un codice errore maggiore e il byte alto è un codice di errore minore.

Cod. Errore Minore	Descrizione	
1 (01h)	Terminato con Successo (questo è il valore di terminazione previsto nella word di stato COMMREQ).	
12 (0Ch)	Errore Locale —Errore di processo comando locale. Il codice di errore minore identifica l’errore specifico.	
	1 (01h)	Comando tipo Attesa non ammesso. Usare il comando Non Attesa.
	2 (02h)	Comando COMMREQ non supportato.
	5 (05h)	Errore di scrittura della word di stato COMMREQ nella memoria PLC.
	6 (06h)	Specificata tipo di memoria PLC non valida.
	7 (07h)	Specificato offset di memoria PLC non valido.
	8 (08h)	Impossibile accedere alla memoria PLC.
	9 (09h)	Superata la lunghezza dati.
	12 (0Ch)	Lunghezza blocco dati COMMREQ insufficiente.
	14 (0Eh)	Dati COMMREQ non validi
	15 (0Fh)	Impossibile allocare le risorse del sistema per completare COMMREQ.
13 (0Dh)	Errore remoto — Errore di processo comando remoto. Il codice errori minore identifica l’errore specifico.	
	2 (02h)	La lunghezza della stringa supera la fine del riferimento del tipo di memoria.
	3 (03h)	Lunghezza blocco dati COMMREQ insufficiente. La stringa dati manca o è incompleta.
	4 (04h)	Timeout di ricezione in attesa di una ricezione seriale di dati
	48 (30h)	Timeout output seriale. La porta seriale non ha potuto inviare la stringa. (Possibile mancanza del segnale CTS quando la porta seriale è configurata per il controllo di flusso hardware.)
	50 (32h)	Timeout COMMREQ. Il COMMREQ non è terminato entro il tempo limite di 20 secondi.
14 (0Eh)	Errore di Composizione Automatica — Errore avvenuto durante l tentativo di inviare una stringa di comando a un modem esterno collegato. Il codice errori minore identifica l’errore specifico.	
	1 (01h)	Non in uso.
	2 (02h)	la stringa di comando modem supera il riferimento della fine del tipo di memoria.
	3 (03h)	Blocco Dati COMMREQ insufficiente. La stringa dati manca o è incompleta.
	4 (04h)	Timeout output seriale. La porta seriale non ha potuto inviare il segnale autocomposiz. modem.
	5 (05h)	Il modem non ha risposto. Controllare il modem e il cablaggio.
	6 (06h)	Risposta OCCUPATO dal modem. Il modem non riesce a completare la connessione richiesta. Il modem remoto è già in uso. Ritentare la richiesta di connessione più tardi.
	7 (07h)	Risposta modem NO CARRIER. Il modem non riesce a completare la connessione richiesta. Controllare il modem locale, quello remoto e la linea telefonica.
	8 (08h)	Risposta modem MANCA LINEA. Il modem non riesce a completare la connessione richiesta. Controllare le connessioni del modem e la linea telefonica.
	9 (09h)	Risposta modem ERRORE. Il modem non riesce a completare il comando richiesto. Controllare la stringa di comando del modem e il modem.
	10 (0Ah)	La risposta modem SQUILLO indica che il modem è stato chiamato da n altro modem. Il modem non riesce a completare il comando richiesto. Ritentare il comando modem più tardi.
	11 (0Bh)	Risposta sconosciuta al modem. Il modem non può completare il comando richiesto. Controllare stringa di comando del modem e modem. La risposta dovrebbe essere COLLEGATO o OK.
	50 (32h)	Timeout COMMREQ. COMMREQ non è terminato entro il tempo limite di 20 secondi.

Comandi COMMREQ I/O Seriale

I seguenti COMMREQ vengono utilizzati per implementare l'I/O Seriale:

- COMMREQ locali – non riceve o trasmette dati tramite la porta seriale.
  - Inizializzazione Porta (4300)
  - Impostazione Buffer Input (4301)
  - Flusso Buffer Input (4302)
  - Lettura Stato Porta (4303)
  - Scrittura Controllo Porta (4304)
  - Annulla Operazione (4399)
- COMMREQ remoti – ricezione e/o trasmissione dati tramite porta seriale.
  - Composizione automatica (4400)
  - Scrittura Byte (4401)
  - Lettura Byte (4402)
  - Lettura Stringa (4403)

Sovrapposizione di COMMREQ

Alcuni dei COMMREQ dell'I/O seriale devono terminare l'esecuzione prima che un altro COMMREQ possa essere elaborato. Altri possono essere lasciati in attesa mentre ne vengono eseguiti altri.

I COMMREQ la cui Esecuzione deve Terminare

- Composizione automatica (4400)
- Inizializza Porta (4300)
- Impostazione Buffer Input (4301)
- Flusso Buffer Input (4302)
- Lettura Stato Porta (4303)
- Scrittura Controllo Porta (4304)
- Annulla Operazione (4399)
- Impostazione Porta Seriale (FFF0)

COMMREQ che Possono Stare in Attesa mentre altri sono in Esecuzione

La tabella qui sotto indica se i COMMREQ Scrivi Byte, Leggi Byte e Leggi Stringa possono rimanere in attesa mentre vengono eseguiti altri COMMREQ.

	NUOVO COMMREQ										
COMMREQ al momento in attesa	Composiz Automatica (4400)	Scrivi Byte (4401)	Inizializza Porta(4300)	Imposta Buffer Input (4301)	Flusso Buffer Input (4302)	Leggi Stato Porta (4303)	Scrivi Controllo Porta (4304	Leggi Byte (4402)	Leggi Stringa (4403)	Annulla Operazione (4399)	Imposta Porta Seriale (FFF0)
Scrivi Byte (4401)	No	No	Sì	Sì	Sì	Sì	Sì	Sì	Sì	Sì	No
Leggi Byte (4402)	No	Sì	Sì	No	No	Sì	Sì	No	No	Sì	No
Leggi Stringa (4403)	No	Sì	Sì	No	No	Sì	Sì	No	No	Sì	No

Funzione Inizializzazione Porta (4300)

Questa funzione invia un comando di reset alla porta specificata. Inoltre, annulla anche qualsiasi COMMREQ a momento in esecuzione e scarica il buffer input interno. L'RTS viene disattivato.

Esempio di Blocco Comando per la Funzione Inizializzazione Porta

	VALORE (decimale)	VALORE (esadecimale)	SIGNIFICATO
indirizzo	0001	0001	lunghezza blocco dati
indirizzo +1	0000	0000	Modo NO ATTESA
indirizzo +2	0008	0008	Tipo di memoria word di stato (%R)
indirizzo +3	0000	0000	Indirizzo word di status meno 1 (%R0001)
indirizzo +4	0000	0000	Non in uso
indirizzo +5	0000	0000	Non in uso
indirizzo +6	4300	10CC	Comando inizializzazione porta

Note Operative

Nota: Le parole di status dei COMMREQ che vengono annullati dall'esecuzione di questo comando non vengono aggiornate.

Attenzione: Se questo COMMREQ viene inviato quando un COMMREQ Scrivi Byte (4401) sta trasmettendo una stringa da una porta seriale, la trasmissione viene interrotta. La posizione all'interno della stringa dove la trasmissione viene interrotta non viene determinata. Inoltre, non viene determinato nemmeno il carattere finale ricevuto dal dispositivo al quale la CPU lo sta inviando.

Impostazione della Funzione Buffer Input (4301)

Questa funzione può essere utilizzata per modificare la capacità del buffer di memoria interna, dove i dati input vengono caricati al loro arrivo. Per default, il buffer viene impostato a un massimo di 2K *byte*.

Recupero Dati dal Buffer

I dati possono essere recuperati dal buffer mediante la funzione Leggi Stringa o Leggi Byte. Non sono direttamente accessibili dal programma applicativo.

Se i dati non vengono recuperati tempestivamente dal buffer, alcuni caratteri possono andare persi.

Esempio di Blocco Comando per Impostare le Funzione Buffer Input

	VALORE (decimale)	VALORE (esadecim.)	SIGNIFICATO
indirizzo	0002	0002	Lunghezza blocco dati
indirizzo +1	0000	0000	Modo NO ATTESA
indirizzo +2	0008	0008	Tipo di memoria word di stato (%R)
indirizzo +3	0000	0000	Indirizzo word di stato meno 1 (%R0001)
indirizzo +4	0000	0000	Non in uso
indirizzo +5	0000	0000	Non in uso
indirizzo +6	4301	10CD	Comando impostazione buffer input
indirizzo +7	0064	0040	Lunghezza buffer (in parole)

Note Operative

Non è possibile impostare la lunghezza del buffer a zero. Se come lunghezza del buffer viene immesso zero, le dimensioni del buffer vengono impostate per default a 2 kByte.

Se viene specificata una lunghezza superiore a 2 kByte, viene generato un errore.

Funzione Svuota Buffer Input (4302)

Questa funzione svuota il buffer input da qualsiasi carattere ricevuto dalla porta seriale, ma non ancora recuperato con il comando leggi. Tutti tali caratteri andranno persi.

Esempio di Blocco Comando per la Funzione Svuota Buffer Input

	VALORE (decimale)	VALORE (esadecimale)	SIGNIFICATO
indirizzo	0001	0001	Lunghezza blocco dati
indirizzo +1	0000	0000	Modo NO ATTESA
indirizzo +2	0008	0008	Tipo di memoria word di stato (%R)
indirizzo +3	0000	0000	Indirizzo word di stato meno 1 (%R0001)
indirizzo +4	0000	0000	Non in uso
indirizzo +5	0000	0000	Non in uso
indirizzo +6	4302	10CE	Comando svuota buffer input

Funzione Leggi Stato Porta (4303)

Questa funzione richiama lo stato corrente della porta. Possono essere rilevati i seguenti eventi:

1. Era stata precedentemente iniziata una richiesta leggi e il numero richiesto di caratteri è stato ricevuto ora, oppure è scaduto il timeout specificato.
2. Era stata precedentemente iniziata una richiesta scrivi e la trasmissione del numero di caratteri specificato è completa, oppure è scaduto il timeout.

Lo stato richiamato da questa funzione indica l’evento o gli eventi che sono terminati. Se precedentemente erano stati iniziati un leggi e uno scrivi, si può manifestare contemporaneamente più di una condizione.

Esempio di Blocco Comando per la Funzione Leggi Stato Porta

	VALORE (decimale)	VALORE (esadecimale)	SIGNIFICATO
indirizzo	0003	0003	Lunghezza blocco dati
indirizzo +1	0000	0000	Modo NO ATTESA
indirizzo +2	0008	0008	Word di stato tipo di memoria (%R)
indirizzo +3	0000	0000	Indirizzo word di stato meno 1 (%R0001)
indirizzo +4	0000	0000	Non in uso
indirizzo +5	0000	0000	Non in uso
indirizzo +6	4303	10CF	Comando leggi lo stato della porta
indirizzo +7	0070	0046	Tipo di memoria stato della porta (%I)
indirizzo +8	0000	0000	Offset memoria stato della porta (%I0001)

Stato della Porta

Lo stato della porta è costituito da una word di stato e dal numero di caratteri nel buffer input che non sono stati recuperati dell'applicazione (caratteri ricevuti e disponibili).

word 1	Word stato della porta (vedi sotto)
word 2	Caratteri disponibili nel buffer input

Lo stato della porta può essere:

Bit	Nome	Definizione	Significato	
15	RI	Lettura in corso	Impostato	Richiamo lettura byte o lettura Stringa
			Cancellato	Tempo scaduto precedente lettura byte o stringa, o terminata
14	RS	Lettura eseguita	Impostato	Lettura byte o lettura stringa completata con successo
			Cancellato	Richiamo nuova lettura byte o lettura stringa
13	RT	Time-out lettura	Impostato	Time out di ricezione ricevuto durante lettura byte o lettura stringa
			Cancellato	Richiamo nuova lettura byte o lettura stringa
12	WI	Scrittura in corso	Impostato	Richiamo nuova scrittura byte
			Cancellato	Tempo scaduto per la scrittura byte precedentemente richiamata, oppure annullata o terminata
11	WS	Scrittura eseguita	Impostato	Scrittura byte precedentemente richiamata completata con successo
			Cancellato	Richiamo nuova scrittura byte
10	WT	Time-out Scrittura	Impostato	Time out di trasmissione ricevuto durante la scrittura byte
			Cancellato	Richiamo nuova scrittura byte
9	CA	Caratteri disponibili	Impostato	Il buffer contiene caratteri non letti
			Cancellato	nessun carattere non letto nel buffer
8	OF	Errore di OverFlow	Impostato	Errore di overflow accaduto nel buffer interno della porta seriale
			Cancellato	Richiamo lettura stato porta
7	FE	Errore di Framing	Impostato	Errore di framing accaduto sulla porta seriale
			Cancellato	Richiamo lettura stato porta
6	PE	Errore di Parità	Impostato	Errore di parità accaduto sulla porta seriale
			Cancellato	Richiamo lettura stato porta
5	CT	CTS attivato	Impostato	La linea CTS in linea sulla porta seriale è attiva o la posta seriale non dispone di una linea CTS
			Cancellato	La linea CTS in linea sulla porta seriale non è attiva
4 - 0	U	non in uso, dovrebbe essere a 0		

Funzione di Controllo Scrittura Porta (4304)

Questa funzione forza l'RTS per la porta specificata:

Esempio di Blocco di comando per la Funzione Controllo Scrittura Porta

	VALORE (decimale)	VALORE (esadecimale)	SIGNIFICATO
indirizzo	0002	0002	Lunghezza blocco dati
indirizzo +1	0000	0000	Modo NON ATTESA
indirizzo +2	0008	0008	Word di stato tipo di memoria (%R)
indirizzo +3	0000	0000	Indirizzo word di stato meno 1 (%R0001)
indirizzo +4	0000	0000	Non in uso
indirizzo +5	0000	0000	Non in uso
indirizzo +6	4304	10D0	Comando di controllo scrittura porta
indirizzo +7	xxxx	xxxx	word controllo porta

Word Controllo Porta

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTS	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

La word controllo porta può essere:

15	**RTS**	Stato comandato per l'output **RTS** 1 = attivazione RTS 0 = disattivazione RTS
0-14	**U**	Non in **Uso** (dovrebbe essere zero)

Note Operative

Per la porta CPU 2 (RS-485), il segnale RTS è legato al driver di trasmissione. Pertanto, il controllo RTS dipende dallo stato corrente del driver di trasmissione. Se il driver di trasmissione non è abilitato, l'asserzione RTS con il COMMREQ controllo scrittura porta non comporta l'asserzione RTS sulla linea seriale. Lo stato del driver di trasmissione è controllato dal protocollo e dipende dal modo duplex in uso sulla porta. Per il modo duplex a 2 e a 4 fili il driver di trasmissione è abilitato solo durante la trasmissione. Pertanto, l'RTS sulla porta seriale verrà visto attivo solo sulla porta 2 (configurata per il modo a 2 o a 4 fili) quando vengono trasmessi dati. Per il modo duplex punto a punto, il driver di trasmissione è sempre abilitato. Pertanto, nel modo duplex punto a punto, l'RTS sulla linea seriale rifletterà sempre quanto scelto con il COMMREQ controllo scrittura porta.

Funzione Annulla Commreq (4399)

Questa funzione annulla l’operazione in corso. Può essere utilizzata per annullare tutte le operazioni, le operazioni di lettura o le operazioni di scrittura.

Se è in corso un’operazione di lettura, e nel buffer input vi sono caratteri non elaborati, tali caratteri vengono lasciati nel buffer input, disponibili per successive letture. La porta seriale non viene resettata

Esempio di Blocco Comando per la Funzione Annulla Operazione

	VALORE (decimale)	VALORE (esadecimale)	SIGNIFICATO
indirizzo	0001	0001	Lunghezza blocco dati
indirizzo +1	0000	0000	Modo NON ATTESA
indirizzo +2	0008	0008	Word di stato tipo di memoria (%R)
indirizzo +3	0000	0000	Indirizzo word di stato meno 1 (%R0001)
indirizzo +4	0000	0000	Non in uso
indirizzo +5	0000	0000	Non in uso
indirizzo +6	4399	112F	Comando annulla operazione
indirizzo +7	0001	0001	Tipo di transazione da annullare 1 Tutte le operazioni 2 Operazioni di lettura 3 Operazioni di scrittura

Note Operative

Questa funzione non aggiorna lo stato delle word dei COMMREQ annullati.

Attenzione: Se questo COMMREQ viene inviato sia nel modo annulla tutto che nel modo annulla scrittura quando dei byte COMMREQ Scrivi (4401) stanno trasmettendo una stringa da un posta seriale, la trasmissione viene interrotta. La posizioni all’interno della quale la trasmissione viene interrotta non viene determinata. Inoltre, anche il carattere finale ricevuto dal dispositivo al quale la CPU sta trasmettendo non viene determinato.

Funzione Composizione Automatica (4400)

Questa funzione consente alla CPU VersaMax di comporre automaticamente il numero di un modem e di inviare una stringa di byte specifica.

Per implementare questa funzione, la porta deve essere configurata per un I/O seriale.

Per esempio, l’enunciazione della ricerca può essere implementata da tre comandi che richiedono tre blocchi di comando COMMREQ:

Composizione Automatica: 04400 (1130h) Compone il numero del modem.

Scrivi Byte: 04401 (1131h) specifica una stringa ASCII, lunga da 1 a 250, da inviare dalla porta seriale.

Composizione Automatica: 04400 (1130h) È responsabilità del programma applicativo PLC interrompere la comunicazione telefonica, cosa che viene eseguita riemettendo il comando di composizione automatica e inviando la stringa di comando di riaggancio.

Blocco di Comando Composizione Automatica

Il comando composizione automatica trasmette automaticamente una sequenza Escape secondo la convenzione di Hayes. Se viene utilizzato un modem che non supporta la convenzione di Hayes, per comporre il numero del modem so potrebbe usare un comando scrivi byte.

Qui sotto sono elencati degli esempi di stringhe di comando comunemente usate per modem non compatibili Hayes:

Stringa di Comando	Lunghezza	Funzione
ATDP15035559999<CR>	16 (10h)	Compone il No a imp. 1-503-555-9999
ATDT15035559999<CR>	16 (10h)	Compone il No tonale 1-503-555-9999
ATDT9,15035559999<CR>	18 (12h)	Composizione tonale con linea esterna a impulsi
ATH0<CR>	5 (05h)	Riaggancia la linea telefonica
ATZ <CR>	4 (04h)	Ripristina la configurazione del modem con i valori salvati all'interno

Esempio di Blocco Comando Composizione Automatica

In questo esempio, il blocco comandi COMMREQ compone il numero234-5678 con un modem compatibile Hayes.

Word	Definizione	Valori
1	0009h	Lunghezza blocco dati CUSTOM (stringa di comando inclusa)
2	0000h	Modo NON ATTESA
3	0008h	Word di stato tipo di memoria (%R)
4	0000h	Indirizzo word di stato meno 1 (%R0001)
5	0000h	Non in uso
6	0000h	Non in uso
7	04400 (1130h)	Numero del comando composizione automatica
8	00030 (001Eh)	Timeout risposta modem (30 secondi)
9	0012 (000Ch)	Numero di byte nella stringa di comando
10	5441h	A (41h), T (54h)
11	5444h	D (44h), T (54h)
12	3332h	No. di telefono: 2 (32h), 3 (33h)
13	3534h	4 (34h), 5 (35h)
14	3736h	6 (36h), 7 (37h)
15	0D38h	8 (38h) <CR> (0Dh)

Funzione Scrivi Byte (4401)

Questa operazione può essere utilizzata per trasmettere uno o più caratteri a un dispositivo remoto mediante la porta seriale specificata. Il/i carattere/i da trasmettere deve/devono essere una word riferimento di memoria. Non può/possono essere cambiato/i fino al termine dell'operazione.

Con un singolo richiamo di questa operazione si possono trasmettere fino a 250 caratteri. Lo stato dell'operazione non è completo fino a che tutti i caratteri saranno trasmessi o in caso di timeout (es. se viene usato il controllo di flusso hardware e il dispositivo remoto non abilita la trasmissione).

Esempio di Blocco Comando per la Funzione Scrivi Byte

	VALORE (decimale)	VALORE (esadecimale)	SIGNIFICATO
indirizzo	0006	0006	Lunghezza blocco dati (caratteri da inviare inclusi)
indirizzo +1	0000	0000	Modo NON ATTESA
indirizzo +2	0008	0008	Word di stato tipo di memoria (%R)
indirizzo +3	0000	0000	Indirizzo word di stato meno 1 (%R0001)
indirizzo +4	0000	0000	Non in uso
indirizzo +5	0000	0000	Non in uso
indirizzo +6	4401	1131	Comando scrivi byte
indirizzo +7	0030	001E	Time-out trasmissione (30 secondi). Vedi nota qui sotto.
indirizzo +8	0005	0005	Numero di byte da scrivere
indirizzo +9	25960	6568	'h' (68h), 'e' (65h)
indirizzo +10	27756	6C6C	'l' (6Ch), 'l' (6Ch)
indirizzo +11	0111	006F	'o' (6Fh)

Nonostante in questo esempio vengano usati caratteri ASCII stampabili, non vi sono restrizioni nei valori dei caratteri che possono essere inviati.

Note Operative

Nota: Specificando zero come timeout di trasmissione, il valore timeout viene impostato sul tempo necessario per trasmettere i dati, più 4 secondi.

Attenzione: Se viene inviato un COMMREQ Inizializza Porta (4300) o un COMMREQ Annulla Operazione (4399) nei modi Annulla Tutto o Annulla Scrivi mentre questo COMMREQ sta trasmettendo una stringa da una porta seriale, la trasmissione viene interrotta. La posizioni all'interno della quale la trasmissione viene interrotta non viene determinata. Inoltre, anche il carattere finale ricevuto dal dispositivo al quale la CPU sta trasmettendo non viene determinato.

Funzione Leggi Byte (4402)

Questa funzione fa in modo che uno o più caratteri vengano letti dalla porta specificata. I caratteri vengono letti dal buffer input interno.

Questa funzione richiama sia il numero di caratteri recuperati che il numero di caratteri non elaborati ancora nel buffer input. Se vengono richiesti zero caratteri input, viene richiamato dal buffer input soltanto il numero di caratteri non elaborati.

Se i caratteri disponibili non sono sufficienti a soddisfare la richiesta e viene specificato un valore non zero per il numero di caratteri da leggere, lo stato dell'operazione non sarà completo fino a che: o verranno ricevuti caratteri sufficienti o interverrà il timeout. In entrambe queste condizioni, lo stato della porta indicherà la ragione che ha fatto concludere l'operazione di lettura. La word di stato non viene aggiornata fino al termine dell'operazione di lettura (sia che intervenga il timeout, sia che vengano ricevuti tutti i dati).

Se l'intervallo di timeout è zero, il COMMREQ rimane in attesa fino a che riceverà la quantità di dati richiesta o fino a che verrà annullato.

Se questo COMMREQ fallisse per qualsiasi ragione, nessun dato viene ritornato al buffer. I dati presenti nel buffer vengono conservati e potranno essere recuperati con una successiva richiesta di lettura.

Esempio di Blocco Comando per la Funzione Leggi Byte

	VALORE (decimale)	VALORE (esadecimale)	SIGNIFICATO
indirizzo	0005	0005	Lunghezza blocco dati
indirizzo +1	0000	0000	Modo NON ATTESA
indirizzo +2	0008	0008	Word di stato tipo di memoria (%R)
indirizzo +3	0000	0000	Indirizzo word di stato meno 1 (%R0001)
indirizzo +4	0000	0000	Non in uso
indirizzo +5	0000	0000	Non in uso
indirizzo +6	4402	1132	Comando leggi byte
indirizzo +7	0030	001E	Time-out lettura (30 secondi).
indirizzo +8	0005	0005	Numero di byte da leggere
indirizzo +9	0008	0008	Dati input tipo di memoria (%R).
indirizzo +10	0000	0000	Dati input indirizzo memoria (%R0001)

Formato Dati Richiamati per la Funzione Leggi Byte

I dati richiamati sono costituiti dal numero di caratteri effettivamente letti, dal numero di caratteri ancora disponibili nel buffer input al termine della lettura (se ne rimangono) e dagli effettivi caratteri input.

indirizzo	Numero di caratteri effettivamente letti
indirizzo +1	No. di caratteri ancora disponibili nel buffer input, se ne rimangono
indirizzo +2	Primi due caratteri (il primo carattere è nel byte basso)
indirizzo +3	Terzo e quarto carattere (il terzo carattere è nel byte basso)
indirizzo +n	Caratteri successivi

Nota Operativa

Se nel parametro del tipo di memoria per i dati input viene specificato che sia una memoria tipo word, in caso venisse effettivamente ricevuto un numero dispari di byte, il byte alto dell'ultima word che viene scritta con i dati ricevuti viene impostato a zero.

Funzione Leggi Stringa (4403)

Questa funzione fa in modo che vengano letti i caratteri dalla porta specificata fino a che verrà ricevuto il carattere di terminazione.

Questa funzione richiama sia il numero di caratteri recuperati, sia il numero di caratteri non elaborati ancora nel buffer input. Se vengono richiesti zero caratteri input, viene richiamato dal buffer input soltanto il numero di caratteri non elaborati.

Se il carattere di terminazione non è presente nel buffer input, lo stato dell'operazione non sarà completo o fino a che verrà ricevuto il carattere di terminazione o fino allo scadere del timeout. In entrambe queste condizioni, lo stato della porta indicherà la ragione che ha fatto concludere l'operazione di lettura.

Se l'intervallo di timeout è zero, il COMMREQ rimane in attesa fino a che riceverà la stringa richiesta, terminante con il carattere finale specificato.

Se questo COMMREQ fallisse per qualsiasi ragione, nessun dato viene ritornato al buffer. I dati presenti nel buffer vengono conservati e potranno essere recuperati con una successiva richiesta di lettura

Esempio di Blocco Comando per la Funzione Leggi Stringa

	VALORE (decimale)	VALORE (esadecimale)	SIGNIFICATO
indirizzo	0005	0005	Lunghezza blocco dati
indirizzo +1	0000	0000	Modo NON ATTESA
indirizzo +2	0008	0008	Word di stato tipo di memoria (%R)
indirizzo +3	0000	0000	Indirizzo word di stato meno 1 (%R0001)
indirizzo +4	0000	0000	Non in uso
indirizzo +5	0000	0000	Non in uso
indirizzo +6	4403	1133	Comando leggi stringa
indirizzo +7	0030	001E	Time-out lettura (30 secondi).
indirizzo +8	0013	000D	Carattere di terminazione (a capo): deve essere tra o e 225 (0xFF) compreso
indirizzo +9	0008	0008	Dati input tipo di memoria (%R).
indirizzo +10	0000	0000	Dati input indirizzo memoria (%R0001)

Formato Dati Richiamati per la Funzione Leggi Stringa

I dati richiamati sono costituiti dal numero di caratteri effettivamente letti, dal numero di caratteri ancora disponibili nel buffer input al termine della lettura (se ne rimangono) e dagli effettivi caratteri input.

indirizzo	Numero di caratteri effettivamente letti
indirizzo +1	No. di caratteri ancora disponibili nel buffer input, se ne rimangono
indirizzo +2	Primi due caratteri (il primo carattere è nel byte basso)
indirizzo +3	Terzo e quarto carattere (il terzo carattere è nel byte basso)
indirizzo +n	Caratteri successivi

Nota Operativa

Se nel parametro del tipo di memoria per i dati input viene specificato che sia una memoria tipo word, in caso venisse effettivamente ricevuto un numero dispari di byte, il byte alto dell'ultima word che viene scritta con i dati ricevuti viene impostato a zero.

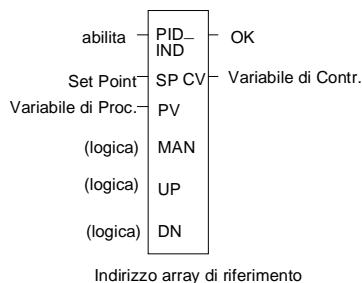
Capitolo 10

La Funzione PID

Questo capitolo descrive la funzione PID (Proporzionale più Integrale più Derivata) che viene utilizzata per controlli di processo in circuito chiuso. Questa funzione PID compara il feedback da un processo variabile con il Set Point di un processo desiderato e aggiorna una Variabile di Controllo basata sull’errore.

- Formato della funzione PID
- Funzionamento della funzione PID
- Blocco Parametro per la funzione PID
- Selezione dell’algoritmo PID
- Definizione delle Caratteristiche del Processo
- Impostazione dei Parametri, Tuning Loop Gain compreso
- Richiamo campione PID

Formato della Funzione PID

La funzione PID utilizza i loop gain PID e altri parametri caricati in un array di 40 word da 16 bit per risolvere l'algoritmo PID all'intervallo di tempo desiderato. Tutti i parametri sono word intere da 16 bit. Ciò consente di usare la memoria %AI per immettere le Variabili di Processo e la memoria %AQ per l'output delle variabili di controllo.

La funzione PID non fa passare corrente se vi è un errore nei parametri configurabili. Può essere controllata con una bobina temporanea durante la modifica dei dati.

Parametri per la funzione PID

Input/ Output	Scelte	Descrizione
abilita	flusso	Quando abilitata da un contatto, viene eseguito l'algoritmo PID.
SP	I, Q, M, T, G, R, AI, AQ, costanti	Il loop di controllo o Set Point del Processo. Impostato usando la conta della variabile di processo, la funzione PID regola l'output delle variabili di controllo, in modo che quelle di processo concordino con il Set Point (zero errori).
PV	I, Q, M, T, G, R, AI, AQ	Input variabili di processo dal processo sotto controllo; spesso un input %AI.
MAN	flusso	Quando attivato a 1 (con un contatto), il blocco PID è in manuale. Se il blocco PID è in manuale off, il blocco PID è in modo autom.
UP	flusso	SE attivato assieme a MAN, regola la variabile di controllo di 1 CV in su a ogni risoluzione.*
DN	flusso	SE attivato assieme a MAN, regola la variabile di controllo di 1 CV in giù ad ogni risoluzione.*
Indirizzo	R	Locazione dell'informazione per il blocco di controllo PID (parametri interni e utente) Usa 40 word %R non condivisibili.
ok	flusso, nessuno	OK viene attivato quando la funzione viene eseguita senza errori. In presenza di errori è off.
CV	I, Q, M, T, G, R, AI, AQ	La variabile di controllo inviata al processo, spesso un output %AQ.

* Incremento (parametro UP) o decremento (parametro DN) di uno (1) per ogni accesso della funzione PID.

Dolo aver scalato16 numeri interi, molti parametri devono essere definiti o nella conta della variabile di processo (PV), o in quella dalla variabile di controllo (CV) o nelle unità. Per esempio, l'input Set Point (SP) deve essere scalato nello stesso range della variabile di processo, in quanto il blocco PID calcola l'errore sottraendo questi due input. Le conte della variabile di processo e della variabile di controllo possono essere da –32000 a 0 o da 0 a 32000, adeguandosi alla scalatura analogica, oppure da 0 a 10000 per visualizzare variabili come da 0,00% a 100,00%. Le conte della variabile di processo e della variabile di controllo non devono necessariamente avere la stessa scalatura, nel qual caso vi saranno fattori di scalatura nei gain PID.

Funzionamento della funzione PID

Funzionamento automatico

La funzione PID può essere richiamata a ogni sweep fornendo corrente al contatto abilita e non fornendo corrente ai contatti input manuale. Il blocco compara il clock per la temporizzazione corrente del PLC con l’ultimo tempo di risoluzione PID caricato nel RefArray interno. Se la differenza è superiore al periodo campione definito nella terza word (%Ref+2) del RefArray, l’algoritmo PID viene risolto usando la differenza di tempo. Sia l’ultimo tempo di risoluzione che l’output della variabile di controllo vengono aggiornati. Nel modo automatico, l’output della Variabile di Controllo viene posto nel parametro Comando Manuale %Ref+13.

Funzionamento manuale

Il blocco PID viene posto ne modo manuale fornendo corrente ai contatti input sia abilita che manuale. L’output per la variabile di controllo viene impostato dal parametro Comando Manuale %Ref+3. Se uno degli input UP o DN sono alimentati, la word comando manuale viene incrementata o decrementata di una conta CV a ogni risoluzione PID. Per cambiamenti manuali più veloci dell’output della variabile di controllo, è anche possibile sommare o sottrarre qualsiasi valore di conta CV direttamente alla o dalla word di comando manuale.

Il controllo PID utilizza i parametri di fermo CV superiore e CV inferiore per limitare l’output CV. Se viene definito un tempo Slew minimo positivo, questo viene usato per limitare la velocità di cambiamento dell’output CV. Se viene superato il limite o di ampiezza o di velocità CV, il valore caricato nell’integratore viene regolato in modo che il CV sia al limite. Questa funzione windup anti-reset significa che anche se l’errore tentasse di portare il CV sopra (o sotto) i fermi per un lungo periodo di tempo, l’output CV escluderà il fermo non appena il termine dell’errore cambia segno.

Questo funzionamento, con il comando manuale che controlla CV nel modo automatico e che imposta CV nel modo manuale, fornisce un passaggio senza inconvenienti tra i modi automatico e manuale. I fermi CV superiore e inferiore e il tempo Slew minimo vengono ancora applicati all’output CV nel modo manuale e il valore interno caricato nell’integratore viene aggiornato. Ciò significa che se si deve passare al comando manuale in modo manuale, l’output CV non potrà cambiare più velocemente della velocità limite del tempo Slew minimo (inverso) e non potrà superare nessuno dei due limiti di fermo CV superiore o inferiore.

Intervallo di tempo per la funzione PID

Il PID non verrà eseguito a una frequenza maggiore di una volta ogni 10 millisecondi. Se viene impostato per essere eseguito a ogni sweep e lo sweep è inferiore a 10 ms, la funzione PID non verrà eseguita fino a che il tempo accumulato dagli sweep eseguiti non avrà superato 10 ms. Per esempio, se il tempo sweep è di 9 ms, la funzione PID verrà eseguita ogni due sweep, pertanto il tempo totale tra due esecuzioni sarà di 18 ms. Una funzione PID specifica non può essere richiamata più di una volta per sweep.

L'intervallo massimo tra due esecuzioni è di 10,9 minuti. La funzione PID compensa il tempo effettivamente trascorso dall'ultima esecuzione entro 100 msec.

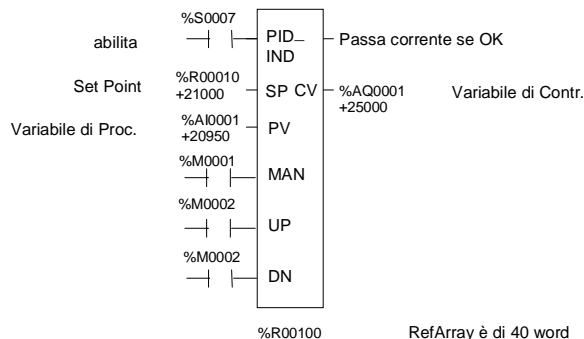
L'algoritmo PID viene risolto solo se il clock di temporizzazione corrente del PLC corrisponde o è superiore al tempo dell'ultima risoluzione PID, più il periodo campione. Se il tempo campione è impostato a 0, la funzione viene eseguita ogni volta che viene abilitata; comunque sempre con la limitazione di 10 ms.

Scalatura input e output

Tutti i parametri della funzione PID sono word intere da 16 bit, perché siano compatibili con le variabili del processo analogico a 16 bit. Alcuni parametri devono essere definiti o in conte PV o in unità, o conte CV o in unità.

Gli input set point devono essere scalati dello stesso range della variabile di processo, perché la funzione PID calcola l'errore sottraendo questi due input. Le conte della variabile di processo e della variabile di controllo non devono necessariamente usare la stessa scalatura. Potrà essere da –32000 a 0 o da 0 a 32000 per corrispondere alla scalatura analogica o da 0 a 10000 per visualizzare variabili da 0,00% a 100,00%. Se la variabile di processo e quella di controllo non usano la stessa scalatura, i fattori di scalatura vengono inclusi nei gain PID.

Esempio della funzione PID

L'esempio che segue comprende input tipici.

Blocco parametro per la Funzione PID

Il blocco parametro per la funzione PID occupa 40 word della memoria %R. Molte delle 40 word vengono usate dal PLC e non sono configurabili. Ogni richiamo delle funzione PID deve utilizzare un'area diversa delle memoria di 40 word, anche se tutti i 13 parametri configurabili sono gli stessi.

Le prime 13 word del blocco parametro devono essere specificate prima di eseguire la funzione PID. Per la maggioranza dei valori di default si possono usare degli zeri. Dopo aver scelto valori PID idonei, questi potranno essere definiti come costanti in un BLKMOV, in modo che possano essere cambiati dal programma quando necessario.

Parametri interni in RefArray

La funzione PID legge 13 parametri e usa le rimanenti 40 word RefArray per memorizzazioni all'interno del PID. Normalmente questi valori non vanno cambiati. Se il blocco PID viene richiamato in modo automatico dopo un lungo ritardo, si potrebbe voler usare SVC_REQ 16 per caricare il clock corrente di temporizzazione in %Ref+23 per aggiornare l'ultimo tempo di risoluzione PID e quindi evitare un cambio step sull'integratore. Se è stato impostato il bit basso di override della word di controllo (%Ref+14) su 1, i quattro bit successivi della word di controllo devono essere impostati per controllare i contatti input del blocco PID e i SP e PV interni devono essere impostati, in quanto il controllo del blocco PID è stato rimosso alla logica ladder.

	Parametro	Unità del Bit basso	Range	Descrizione
indirizzo	Numero Loop	Intero	da 0 a 255.	Numero opzionale del blocco PID. Fornisce un'identificazione comune nel PLC quando il numero loop e definito da un'interfaccia operatore.
indirizzo +1	Algoritmo	-	Impostato dal PLC	1 = algoritmo ISA 2 = algoritmo indipendente
indirizzo +2	Periodo campione	10ms	da 0 (ogni sweep) a 65535 (10.9 Min) Almeno 10ms.	Il tempo più breve, con incrementi di 10 ms, tra le risoluzioni dell'algoritmo PID. Per esempio, usare 10 per un periodo campione di 100 ms.
indirizzo +3 indirizzo +4	Dead Band + e Dead Band -	Conte PV	da 0 a 32000 (+ mai negativo) (- mai positivo)	Valori INT che definiscono i limiti Dead Band superiore (+) e inferiore (-) nelle conte PV. Se la dead band non serve, questi valori devono essere 0. Se l'errore PID (SP – PV) 0 (PV – SP) supera il valore (-) o è inferiore al valore (+), i calcoli PID vengono risolti con un errore di 0. Se non zero, il valore (+) deve essere superiore a 0 e il valore (-) inferiore a 0, altrimenti il blocco PID non funziona. Lasciarli a 0 fino a che vengono impostati o regolati i gain PID loop. Aggiungere una deadband per evitare piccoli cambiamenti output CV dovuti a variazioni dell'errore.

	Parametro	Unità del Bit basso	Range	Descrizione
indirizzo +5	Gain proporzional e -Kp (Controller gain, Kc, nella versione ISA)	0.01 CV%/PV%	da 0 a 327.67%	Cambio nella variabile di controllo nelle conte CV per un cambio conta 100 PV nel termine di Errore. Un Kp immesso come 450 viene visualizzato come 4.50 e risulta in un contributo Kp*Errore/100 o 450*Errore/100 per l'output PID. Kp è generalmente il primo gain impostato quando si regola il loop PID.
indirizzo +6	Gain-Kd derivato	0.01 secondi	da 0 a 327.67 sec	Cambio nella variabile di controllo nelle conte CV se l'errore o il PV cambiano di 1 conta PV ogni 10ms. Immesso come tempo con il bit basso che indica 10ms. Per esempio, un Kd immesso come 120 viene visualizzato come 1,20 sec e risulta in un contributo di tempo Kd* delta o errore/delta o di 120*4/3 per l'output PID se l'errore cambiava di 4 conte PV ogni 30 ms. Kd può essere usato per accelerare una risposta loop lenta, ma è molto sensibile al disturbo input PV.
indirizzo +7	Rate-Ki integrale	Ripetizione/ 1000 Sec	da 0 a 32.767 ripetizione/sec	Cambio nella variabile di controllo nelle conte CV se l'errore era una conta costante di 1 PV. Visualizzato come 0.000 ripetizioni/sec con una virgola decimale implicita di 3. Per esempio, un Ki immesso come 1400 viene visualizzato come 1,400 ripetizioni/sec e risulta in un contributo Ki * Error*dt o 1400 * 20 * 50/1000 all'output PID per un errore di 20 conte PV e un tempo sweep PLC di 50 ms (periodo campione 0). Ki e di solito la seconda impostazione gain dopo Kp.
indirizzo +8	CV Bias/Output Offset	Conta CV	da -32000 a 32000 (sommati all'output integratore)	Numero di conte CV aggiunto all'output PID prima dei fermi di velocità e ampiezza. Può essere usato per impostare valori CV non zero se vengono usati solo gain PV proporzionali e per passare oltre il controllo di questo PID output loop da un altro controllo loop.
indirizzo +9 indirizzo+10	Fermi sup. e inf. CV	Conta CV	-32000 a 32000 (>%Ref+10)	Numero di conte CV che definiscono il valore CV più alto e più basso. Questi valori sono necessari. Il fermo superiore deve avere un valore più positivo di quello inferiore, altrimenti il blocco PID non funziona. Questi vengono generalmente usati per definire limiti basati sui limiti fisici di un output CV. Vengono usati anche per scalare il display del grafico a barre CV. Il blocco ha un windup anti-reset per modificare il valore dell'integratore quando viene raggiunto un fremo CV.
indirizzo+11	Tempo Slew minimo	Secondi/corsa intera	da 0 (nessuno) a 32000 sec per passare a CV 32000	Numero minimo di secondi per il passaggio dell'output CV da 0 al fine corsa del 100% o 32000 conte CV. Si tratta di un limite inverso di velocità per la velocità alla quale può cambiare l'output CV. Se positivo, CV non può cambiare più di 32000 conte CV per il tempo Delta (secondi) diviso per il tempo Slew minimo. Per esempio, se il periodo campione è di 2,5 sec. e il tempo minimo Slew e di 500 secondi, il CV non può cambiare più di 32000*2,5/500 o 160 conte CV per risoluzione PID. Il valore dell'integratore viene regolato se viene superata la velocità CV limite. Impostare il tempo minimo Slew a 0 mentre si sintonizzano o si regolano i gain PID loop.

	Parametro	Unità del Bit basso	Range	Descrizione
indirizzo+12	Word di configurazione	Vengono usati 5 bit bassi	Bit da 0 a 2 per Errore+/-, Polarità output, Derivata.	I 5 bit bassi di questa word vengono usati per modificare tre impostazioni PID standard. Gli altri bit vanno impostati a 0. Impostare il bit basso a 1 per modificare il termine standard errore PID da normale (SP – PV) a (PV – SP), invertendo il segno del termine di ritorno. Ciò serve per i controlli ad azione invertita, dove il CV si deve abbassare quando il PV aumenta. Impostare il secondo bit a 1 per invertire la polarità output, in modo che CV sia l'output negativo del PID invece di avere il normale valore positivo. Impostare il quarto bit a 1 per modificare l'azione derivata dall'uso del cambiamento normale nel termine di errore all'uso del cambiamento del termine di ritorno PV. Dettagli dei 5 bit bassi della word Config: Bit 0: Termine Errore. Quando questo bit è 0, il termine errore è SP – PV; quando è 1 il termine errore è PV - SP. Bit 1: Polarità Output. Quando questo bit è 0, l'output CV rappresenta l'output del calcolo PID. Quando è 1, l'output CV rappresenta il negativo dell'output del calcolo PID. Bit 2: Azione derivata su PV. Quando questo bit è 0, l'azione derivata viene applicata al termine d'errore. Quando è 1 l'azione derivata viene applicata al PV. I rimanenti bit dovrebbero essere a 0. Bit 3: Azione Deadband. Quando il bit dell'azione Deadband è 0, non viene scelta lacuna azione deadband. Se l'errore rientra nei limiti deadband, l'errore vede essere zero, diversamente l'errore non viene influenzato dai limiti deadband. Se il bit azione deadband è 1, l'azione deadband viene scelta. Se l'errore rientra nei limiti deadband, l'errore viene forzato a essere zero. Se però l'errore sconfina dai limiti deadband, l'errore viene ridotto dal limite deadband (error = errore limite deadband). Bit 4: Azione windup Anti-reset. Quando questo bit è 0, l'azione windup anti-reset usa un calcolo reset back. Quando l'output è al fermo, questo sostituisce il valore remanider Y accumulato con qualsiasi valore fosse necessario per produrre esattamente l'output fermato. Quando questo bit è 1, questo sostituisce il termine Y accumulato con il valore del termine Y all'inizio del calcolo. In questo modo, il valore Y pre-fermo viene conservato fino a che l'output viene fermato. Si ricordi che questi bit vengono impostati in potenze di 2. Per esempio, per impostare la word Config a 0 per la configurazione PID di default, si deve aggiungere un 1 per cambiare il termine errore da SP-PV a PV-SP o aggiungere 2 per cambiare la polarità output da CV = output PID a CV = -output PID o aggiungere 4 per cambiare l'azione derivata da velocità di errore del cambiamento a velocità di cambiamento PV, ecc.
indirizzo+13	Comando Manuale	Conte CV	Controlla CV in Auto o imposta CV in manuale	Imposta l'output CV corrente mentre il blocco PID è nel modo automatico. Quando il blocco viene commutato nel modo manuale, questo valore viene usato per impostare l'output CV e il valore interno dell'integratore tra i limiti superiore e inferiore di fermo e i limiti del tempo Slew.

<table>
<tr><th></th><th>Parametro</th><th>Unità del Bit basso</th><th>Range</th><th>Descrizione</th></tr>
<tr><td>indirizzo+14</td><td>Word di Controllo</td><td>Mantenuta dal PLC, se non è impostato il Bit 1.</td><td>Mantenuto dal PLC se non diversamente impostato; il bit basso imposta l’override se 1.</td><td>Se il bit basso di override è su 1, questa word e gli altri parametri interni SP, PV e CV devono essere usati per il funzionamento remoto di questo blocco PID (vedere sotto). Ciò consente a un’interfaccia operatore remota, es. un computer, di rilevare il controllo dal programma PLC. Attenzione: se non si vuole che ciò accada, verificare che la word di controllo sia su 0. Se il bit basso è 0, si possono leggere i 4 bit successivi per seguire lo stato dei contatti input del PID fino a che il contatto abilita PID ha corrente.
Una struttura discreta dati con le posizioni dei primi cinque bit nel seguente formato:
<table>
<tr><td>Bit:</td><td>Valore Word:</td><td>Funzione:</td><td>Stato dell’azione esterna se il bit override è impostato su 1:</td></tr>
<tr><td>0</td><td>1</td><td>Override</td><td>Se 0, il monitor blocca i contatti qui sotto. Se 1, li imposta esternamente.</td></tr>
<tr><td>1</td><td>2</td><td>Manuale /Auto</td><td>Se 1, il blocco è in modo manuale, con altri numeri è in modo auto.</td></tr>
<tr><td>2</td><td>4</td><td>Abilita</td><td>Normalmente deve essere 1, diversamente blocco mai richiamato.</td></tr>
<tr><td>3</td><td>8</td><td>UP /Aumenta</td><td>Se 1e il manuale (Bit 1) è 1, CV incrementa a ogni risoluzione.</td></tr>
<tr><td>4</td><td>16</td><td>DN /Dim.</td><td>Se 1 e il manuale (Bit 1) è 1, CV incrementa a ogni risoluzione.</td></tr>
</table></td></tr>
<tr><td>indirizzo+15</td><td>SP interno</td><td>Impostato e mantenuto dal PLC</td><td>Non-configu-rabile</td><td>Controlla SP in; deve essere impostato esternamente se Override = 1.</td></tr>
<tr><td>indirizzo+16</td><td>CV interno</td><td>“</td><td>“</td><td>Controlla CV out.</td></tr>
<tr><td>indirizzo+17</td><td>PV interno</td><td>“</td><td>“</td><td>Controlla PV in; deve essere impostato esternamente se Override bit = 1.</td></tr>
<tr><td>indirizzo+18</td><td>Output</td><td>“</td><td>“</td><td>Valore word con segno che rappresenta l’output del blocco funzione prima dell’inversione opzionale. Se non viene configurata l’inversione output e il bit di polarità output nella word di controllo è su 0, questo valore è pari all’output CV. Se viene selezionata l’inversione e il bit di polarità output è 1, questo valore è pari al negativo dell’output CV.</td></tr>
<tr><td>indirizzo+19</td><td>Caricam. Diff Term</td><td></td><td></td><td></td></tr>
<tr><td>indirizzo+20
indirizzo+21</td><td>Caricam Int Term</td><td></td><td></td><td>Usato internamente per caricare valori intermedi. Non scrivere in queste locazioni.</td></tr>
<tr><td>indirizzo+22</td><td>Caricam. Slew Term</td><td></td><td></td><td></td></tr>
<tr><td>Da indirizzo +23 a indirizzo +25</td><td>Clock</td><td></td><td></td><td>Caricamento temporizzazione interna (dell’ultimo PID eseguito). Non scrivere in queste locazioni.</td></tr>
<tr><td>indirizzo+26</td><td>Caricamento Remainder Y</td><td></td><td></td><td>Conserva il reminder per la divisione a scalare dell’integratore perché lo stato errore sia sempre 0.</td></tr>
<tr><td>indirizzo+27
indirizzo+28</td><td>Range SP, PV sup. e inf.</td><td>Conte PV</td><td>da -32000 a 32000</td><td>Valori opzionali INT in conte PV che definiscono i valori visualizzati alti e bassi (Ref +27 deve essere inf. a Ref+28)</td></tr>
</table>

	Parametro	Unità del Bit basso	Range	Descrizione
Da indirizzo+29 a indirizzo+39	Riservato	N/A	Non-configu-rabile	29-34 riservati per uso interno; 35-39 riservati per uso esterno. Non usare questi riferimenti.

Selezione algoritmo PID (PIDISA o PIDIND) e dei Gain

Il blocco PID può essere programmato selezionando o il termine indipendente (PID_IND) o le versioni standard ISA (PID_ISA) dell'algoritmo PID. La sola differenza nell'algoritmo è come vengono definiti i gain Interi e Derivati.

Entrambi i tipi di PID calcolano il termine errore come SP – PV, che può essere cambiato ne modo azione inversa PV – SP impostando il termine errore (bit basso 0 nella word Config. %Ref+12) su 1.

Il modo azione inversa può essere usato se si vuole che l'output CV vada nella direzione inversa dei cambiamenti input PV (CV giù per PV su) invece che nel modo normale CV su per PV su.

Errore = (SP - PV)   o (PV - SP) se il bit basso della word Config è su 1

La derivata è normalmente basata sul cambio del termine di errore dopo l'ultima risoluzione PID, che può causare grosse variazioni nell'output se il valore SP viene cambiato. Diversamente, il terzo bit della word Config può essere impostato su 1, per calcolare la derivata basata sui cambiamenti PV. Il dt (o tempo delta) viene determinato sottraendo il tempo dell'ultima risoluzione PID per questo blocco dalla temporizzazione PLC corrente.

dt = Temporizzazione PLC corrente – Tempo dell'ultima risoluzione PID
Derivata = (Errore – errore prec.)/dt o (PV – PV prec.)/dt se il 3o bit della word Config è su 1

L'algoritmo termine indipendente PID (PID_IND) calcola l'output come:

PID Output = Kp * Errore +   Ki * Errore * dt + Kd * Derivata + Bias CV

L'algoritmo standard ISA (PID_ISA) ha una formato diverso:

PID Output = Kc * (Errore +   Errore * dt/Ti + Td * Derivata) + Bias CV

dove Kc è il gain controller e Ti è il tempo intero e Td è il tempo derivato. Il vantaggio di ISA è che regolando i cambiamenti Kc cambiano il contributo dei termini interi e derivati nonché di quello proporzionale, che potrebbero facilitare la fasatura del loop. Se i gain PID sono in termini o Ti e Td, usare

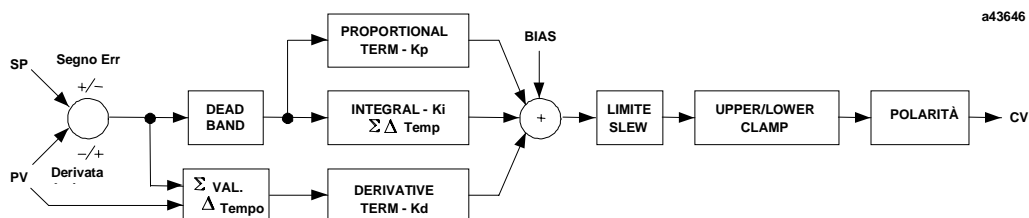
Kp = Kc   Ki = Kc/Ti e Kd = Kc/Td

per convertirli a usarli come PID i parametri input dell'utente.

Il termine Bias CV è un termine aggiuntivo separato dai componenti PID. Potrebbe rendersi necessario se si usa solo il gain Kp proporzionale e si viole che CV sia un valore non zero quando il PV è pari a SP e l'errore è 0. In questo caso, impostare il Bias CV al CV desiderato quando il PV è al SP. Il Bias CV può essere usato anche per far avanzare il controllo, dove un altro loop PID o un algoritmo di controllo viene usato per regolare l'output CV di questo loop PID.

Se viene usato un gain Ki intero, il Bias CV dovrebbe normalmente essere 0, in quanto l'integratore agisce come bias automatico. Basterà iniziare nel modo manuale e usare la word di comando manuale (%Ref+13) per impostare l'integratore sul CV desiderato, quindi passare la modo automatico. Ciò vale anche se Ki è 0, con l'eccezione che l'integratore non verrò regolato in base all'errore dopo essere passati al modo automatico.

Algoritmo termine indipendente (PIDIND)

Lo schema qui sotto mostra come opera l'algoritmo PID:

L'algoritmo ISA (PIDISA) è simile, con l'eccezione che il gain Kp è un fattore all'esterno di Ki e Kp, in modo che il gain integrale è Kp * Ki e il gain derivato è Kp * Kd. Il segno di errore, l'azione derivata e la polarità sono impostati da bit nel parametro utente della word Config.

Ampiezza CV e limiti di velocità

Il blocco non invia l’output PID calcolato direttamente alla CV. Entrambi gli algoritmi PID possono imporre l’ampiezza e la velocità dei limiti di cambiamento sulla variabile di controllo output. La velocità massima di cambiamento viene determinata controllando il massimo valore 100% CV (32000) con il tempo minimo Slew, se specificato superiore a 0. Per esempio, se il tempo minimo Slew è 100 secondi, la velocità limite sarà 320 conte CV al secondo. Se il tempo di risoluzione è stato di 50 millisecondi, il nuovo output CV non potrà cambiare più di 320*50/1000 o 16 conte CV rispetto al precedente output CV.

L’output CV viene poi comparato ai valori di fermo CV superiore e inferiore. Se viene superato uno dei limiti, l’output CV viene impostato al valore del fermo. Se i limiti di velocità o di ampiezza vengono superati modificando il CV, il valore dell’integratore interno viene regolato in modo che risponda al valore limite ed eviti il reset windup.

In fine, il blocco controlla la polarità dell’output (2° bit della word Config %Ref+12) e cambia il segno dell’output se il bit è 1.

CV = Fermo output PID o
- Fermo output PID se è impostato il bit di polarità output

Se il blocco è nel modo automatico, il CV finale viene posto ne comando manuale %Ref+13. Se il blocco è nel modo manuale, l’equazione PID viene saltata, in quanto la CV viene impostata dal comando manuale, ma vengono ancora controllati tutti i limiti di velocità e ampiezza. Ciò significa che il comando manuale non può cambiare l’output oltre i fermi CV superiore e inferiore e che l’output non può cambiare più velocemente del tempo minimo Slew ammesso.

Periodo campione e programmazione del blocco PID

Il blocco PID è un'implementazione digitale di una funzione di controllo analogica, pertanto il tempo campione dt nell'equazione output PID non è il tempo campione infinitesimalmente piccolo disponibile con i controlli analogici. La maggioranza dei processi che vengono controllati possono essere approssimati come gain con un ritardo del primo o secondo ordine, possibilmente con un tempo di ritardo puro. Il blocco PID imposta un output CV per il processo e usa il PV di feedback del processo per determinare un errore che regola il successivo output CV. Un parametro chiave del processo è il tempo totale costante, cioè quanto velocemente risponde il PV quando il CV viene cambiato. Come descritto nel paragrafo Impostazione dei Gain Loop, il tempo totale costante, Tp+Tc, per un sistema di primo ordine è il tempo necessario a PV per raggiungere il 63% del suo valore finale quando CV incrementa. Il blocco PID non è in gradi do controllare un processo, a meno che il periodo campione sia decisamente inferiore alla metà del tempo totale costante. Tempi campione maggiori lo renderebbero instabile

Il periodo campione non dovrebbe essere superiore al tempo totale costante diviso per 10 (o fino a 5 nel peggiore dei casi). Per esempio, se il PV pare che raggiunga i 2/3 del suo valore finale in 2 secondi, il periodo campione dovrebbe essere inferiore a 0,2 secondi, o a 0,4 secondi nel peggiore dei casi. D'altra parte, il periodo campione non dovrebbe essere troppo breve, cioè inferiore al tempo totale costante diviso per 1000, altrimenti il termine Ki * Errore * dt per l'integratore PID lo arrotonderebbe a 0. Per esempio, un processo lentissimo che impiega 10 ore o 36000 secondi per raggiungere il livello del 63% dovrebbe avere un periodo campione di 40 secondi o più lungo.

A meno che il processo non sia molto veloce, normalmente non è necessario usare un periodo campione di 0 per risolvere l'algoritmo PID a ogni sweep PID. Se vengono usati molti loop PID con un periodo campione superiore al tempo sweep, potrebbero verificarsi ampie variazioni nel tempo sweep PLC se molti loop terminano risolvendo contemporaneamente l'algoritmo. La soluzione più semplice consiste nel sequenziare uno o più bit 1 lungo un array di bit impostati su 0, che vengono usati per abilitare il passaggio di corrente a singoli blocchi PID.

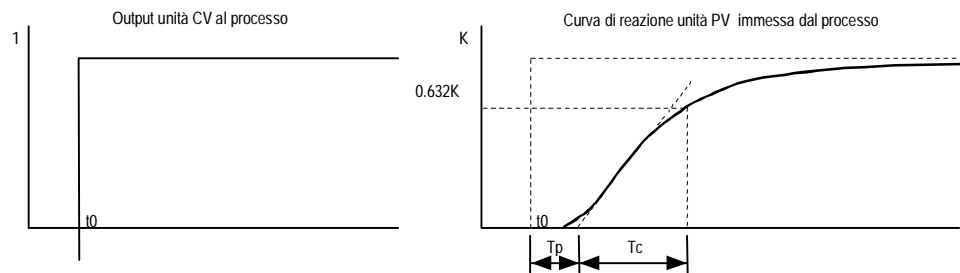
Determinazione delle Caratteristiche del Processo

I gain PID loop, Kp, Ki e Kd, vengono determinati dalle caratteristiche del processo da controllare. Le due domande nell'impostazione di un PID loop sono:

1. Di quanto cambia il PV quando viene modificato il CV di una quantità fissa, oppure, quanto è il gain open loop?
2. Quanto velocemente risponde il sistema, oppure, quanto rapidamente cambia il PV dopo che l'output CV viene incrementato?

Molti processi possono essere approssimati con in gain del processo, con ritardi del primo o secondo ordine e un ritardo in tempo puro. In termini di frequenza, la funzione di trasferimento per un sistema a ritardo del primo ordine con un ritardo in tempo puro è:

PV(s)/CV(s) = G(s) = K * e **(-Tp s)/(1 + Tc s)

Tracciando una risposta all'incremento al tempo t0 in termini di tempo si fornisce una curva di reazione dell'unità open loop:

I parametri del seguente modello di processo possono essere determinati dalla curva di reazione dell'unità PV:

K	Gain open loop del processo = cambiamento finale in PV/cambiamento in CV al tempo t0 (Si noti K senza alcun deponente)
Tp	Ritardo del processo o delle tubazioni o tempo morto dopo t0 prima che l'output PV del processo inizi a spostarsi
Tc	Costante di tempo del primo ordine del processo, tempo necessario dopo Tp perché PV raggiunga il 63% della PV finale

Generalmente, il modo più veloce per misurare questi parametri sta ne porre il blocco PID nel modo manuale per produrre un piccolo incremento nell'output CV, modificando il comando manuale %Ref+13 e tracciando la risposta PV nel tempo. Per processi lenti, ciò può essere eseguito manualmente, ma per processi più veloci, sarà utile un registratore a carta o un pacchetto software per il trattamento dati. L'intensità dell'incremento CV dovrebbe essere abbastanza ampio da provocare una variazione rilevabile in PV, ma non tale da disturbare il processo sotto controllo. Un valore adatto potrebbe essere tra il 2% e il 10% di differenza tra i valori di fermo CV superiore e inferiore.

Impostazione dei Parametri Compresi i Gain Tuning Loop

Tutti i parametri PID dipendono completamente da processo da controllare e non vi sono valori predeterminati validi; generalmente è abbastanza semplice individuare un loop gain accettabile.

1. Azzerare tutti i parametri utente, quindi impostare i fermi CV superiore e inferiore alla CV massima e minima prevista. Impostare il periodo campione sulla costante(superiore)/da 10 a 100 di tempo stimata per il processo.
2. Porre il blocco nel modo manuale e impostare il comando manuale (%Ref+13) a valori diversi per controllare se la CV può essere portata ai fermi superiore e inferiore. Annotare il valore PV a qualche punto CV e caricarlo in SP.
3. Impostare un piccolo gain, tipo 100 * CV massimo/PV massimo, il Kp ed escludere il modo manuale. Impostare SP dal 2% al 10% del massimo range PV e osservare la risposta PV. Se la risposta di incremento PV è troppo lenta, aumentare Kp, oppure ridurlo se PV è eccessiva e oscilla senza raggiungere un valore stabile.
4. Una volta determinato Kp, iniziare ad aumentare Ki fino a sconfinare e poi ridurlo a un valore stabile in 2 o 3 cicli. Ciò potrebbe comportare la riduzione di Kp. Tentare anche con incrementi diversi e con punti operativi CV.
5. Dopo aver trovato i gain Kp e Ki idonei, tentare aggiungendo Kd per avere una risposta più rapida alle variazioni input, curando che non provochi oscillazioni. Spesso Kd non è necessario e non va bene con PV disturbate.
6. Controllare i gain con diversi punti operativi SP e, se necessario, aggiungere Deadband e il tempo minimo Slew. Con alcuni processi ad azione inversa potrebbe essere necessario impostare il segno per la word Config di errore o i bit di polarità.

Impostazione dei gain loop con l'approccio di fasatura Zeigler e Nichols

Una volta determinati i tre parametri del modello di processo, K, Tp e Tc, potranno essere usati per valutare i loop gain PID iniziali. Il seguente approccio fornisce una buona risposta ai disturbi del sistema con gain che producono un rapporto di ampiezza di 1/4. Il rapporto di ampiezza è il rapporto tra il secondo e il primo picco nella risposta a circuito chiuso.

1. Calcolare la velocità di reazione:

   R = K/Tc

2. Soltanto per il controllo proporzionale, calcolare Kp come:

   Kp = 1/(R * Tp) = Tc/(K * Tp)

   Per il controllo proporzionale e integrale, usare:

   Kp = 0.9/(R * Tp) = 0.9 * Tc/(K * Tp) Ki = 0.3 * Kp/Tp

   Per il controllo proporzionale, integrale e derivato, usare:

   Kp = G/(R * Tp)   dove G va da 1.2 a 2.0
   Ki = 0.5 * Kp/Tp
   Kd = 0.5 * Kp * Tp

3. Controllare che il periodo campione rientri nel range

   da (Tp + Tc)/10 a (Tp + Tc)/1000

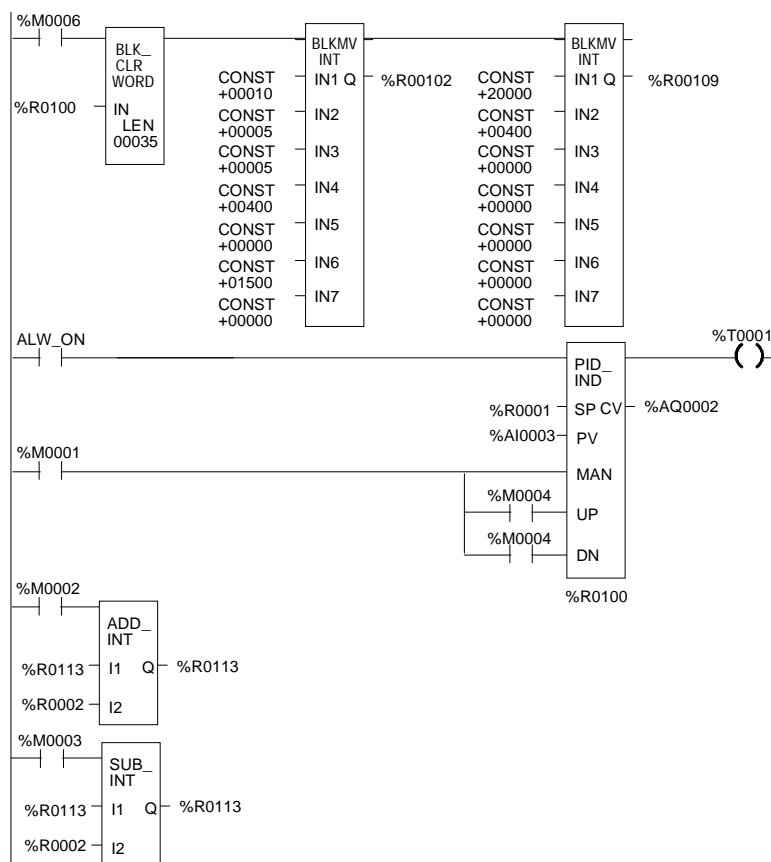
Il metodo ideale di fasatura

la procedura "Fasatura Ideale" fornisce la miglior risposta alle variazioni SP, ritardate soltanto dal ritardo del processo Tp o dai tempi morti.

Kp = 2 * Tc/(3 * K * Tp)
Ki = Tc
Kd = Ki/4   se viene usato il termine derivato

Una volta determinati i gain iniziali, convertirli in integrali. Calcolare il gain del processo K come la variazioni delle conte input PV divise per la variazione dell'incremento output nelle conte CV e non nelle unità di processo tecniche PV o CV. Specificare tutti i tempi in secondi. Una volta determinati Kp, Ki e Kd, Kp e Kd possono essere moltiplicati per 100 e immessi come integrali, mentre Ki può essere moltiplicato per 1000 e immesso nel parametro utente %RefArray.

PID campione

I seguenti esempi PID hanno un periodo campione di 100ms, un gain Kp di 4.00 e un gain Ki di 1.500. Il set point è caricato in %R0001, l'output variabile di controllo in %AQ0002, e la variabile di processo è inviata a %AI0003. Si devono impostare i fermi CV superiore e inferiore, in questo caso a 20000 e 4000, ed è stata inclusa una piccola Dead Band opzionale di +5 e -5. Il RefArray di 40 word inizia in %R0100. Normalmente i parametri utente vengono impostati in RefArray, ma si può impostare %M0006 per reinizializzare le 14 word iniziando a %R0102 (%Ref+2) dalle costanti caricate nella logica (una tecnica utile).

Il blocco può essere commutato in modo manuale con %M1 in modo che il comando manuale, %R113, possa essere regolato. I bit %M4 o %M5 possono essere usati per aumentare o diminuire %R113 e la CV PID e l'integratore di 1 a ogni risoluzione in 100 Msec. Per un funzionamento manuale più veloce, i bit %M2 e %M3 possono essere usati per sottrarre il valore in %R2 a/da %R113 a ogni sweep PLC. L'output %T1 è on quando il PID è OK.

Appendice A Temporizzazione

Questa appendice riporta informazioni sulla temporizzazione della CPU per ciascuna delle funzioni disponibili nel set di istruzioni CPU. Il tempo di abilitazione vale per unità a lunghezza singola del tipo %R.

Informazioni di temporizzazione da fornire.

Per le funzione della tabella, vengono specificai incrementi in unità di lunghezza: per funzioni mediante bit, microsecondi/bit; per funzioni movimento dati, microsecondi/numero di bit o word

Tempo di abilitazione per unità di lunghezza singola del tipo %R, %AI, e %AQ.

DOIO è il tempo di invio di valori a moduli output discreti.

Dove i casi possibili sono più di uno, il tempo indicato rappresenta il caso peggiore.

Per istruzioni con valore incrementale, moltiplicare l'incremento per (lunghezza – 1) e aggiungere tale valore al tempo base.

Tempi di esecuzione

I tempi di esecuzione sono indicati per ciascuna funzione:

Abilitato	Tempo richiesto per eseguire la funzione o il blocco funzione quando nella funzione passa corrente. Tipicamente, i tempi dei casi migliori si hanno quando i dati utilizzati dal blocco sono contenuti nella RAM utente (memoria orientata word) e non nella memoria cache ISCP (memoria discreta).
Disabilitato	Tempo richiesto per eseguire la funzione quando nella funzione o nel blocco funzione passa corrente, ma si trovano in stati di inattività, come quando un timer si trova in stato di reset.

Nota

I timer e i contatori vengono aggiornati ogni volta che vengono rilevati nella logica: i timer con la quantità di tempo impiegato dall'ultimo sweep, i contatori ci una conta.

Timer, contatori, funzioni matematiche, funzioni trig, funzioni log

Gruppo	Funzione	Abilitata	Disabilitata	Incremento	Quantità
Timer	Timer ritardo ON	–	–	–	15
	Timer	–	–	–	15
	Timer ritardo OFF	–	–	–	15
Contatori	Contatore su	–	–	–	13
	Contatore giù	–	–	–	13
Matematica	Addizione (INT)	–	–	–	13
	Addizione (DINT)	–	–	–	19
	Addizione (REALE)	–	–	–	17
	Sottrazione (INT)	–	–	–	13
	Sottrazione (DINT)	–	–	–	19
	Sottrazione (REALE)	–	–	–	17
	Moltiplicazione (INT)	–	–	–	13
	Moltiplicazione (DINT)	–	–	–	19
	Moltiplicazione (REALE),	–	–	–	17
	Divisione (INT)	–	–	–	13
	Divisione (DINT),	–	–	–	19
	Divisione (REALE)	–	–	–	17
	Modulo Divisione (INT)	–	–	–	13
	Modulo Div (DINT)	–	–	–	19
	Radice quadrata (INT)	–	–	–	10
	Radice quadrata (DINT)	–	–	–	13
	Radice quadrata (REALE)	–	–	–	11
Trigonometrica	SIN (REALE)	–	–	–	11
	COS (REALE)	–	–	–	11
	TAN (REALE)	–	–	–	11
	ASIN (REALE)	–	–	–	11
	ACOS (REALE)	–	–	–	11
	ATAN (REALE)	–	–	–	11
Logaritmica	LOG (REALE	–	–	–	11
	LN (REALE	–	–	–	11

Funzioni esponenziali, conversione radianti, funzioni relazionali

Gruppo	Funzione	Abilitata	Disabilitata	Incremento	Quantità
Esponenziale	EXP,	–	–	–	11
	EXPT	–	–	–	17
Radiante	Converti RAD in DEG	–	–	–	11
Conversione	Converti DEG in RAD				11
Relazionale	Uguale (INT)	–	–	–	10
	Uguale l (DINT)	–	–	–	16
	Uguale (REALE)	–	–	–	14
	Non uguale (INT)	–	–	–	10
	Non uguale (DINT)	–	–	–	16
	Non uguale (REALE)	–	–	–	14
	Maggiore di (INT)	–	–	–	10
	Maggiore di (DINT)	–	–	–	16
	Maggiore di (REALE)	–	–	–	14
	Maggiore di/uguale (INT)	–	–	–	10
	Maggiore di/uguale (DINT)	–	–	–	10
	Maggiore di/uguale (REALE)	–	–	–	14
	Minore di (INT)	–	–	–	10
	Minore di (DINT)	–	–	–	16
	Minore di (REALE)	–	–	–	14
	Minore di/uguale (INT)	–	–	–	10
	Minore di/uguale (DINT)	–	–	–	16
	Minore di/uguale (REALE)	–	–	–	14
	Range (INT)	–	–	–	13
	Range (DINT)	–	–	–	22
	Range (WORD)	–	–	–	13

Operazioni Bit, funzioni spostamento dati

Gruppo	Funzione	Abilitata	Disabilitata	Incrementi	Quantità
Bit Operation	Logico AND	–	–	–	13
	Logico OR	–	–	–	13
	Logico Esclusivo OR	–	–	–	13
	Logico Inversione, NOT				10
	sposta Bit a sinistra	–	–	–	16
	Sposta Bit a destra	–	–	–	16
	Ruota Bit a sinistra	–	–	–	16
	Ruota Bit a destra	–	–	–	16
	Posizione Bit	–	–	–	13
	Cancella Bit	–	–	–	13
	Bit Test	–	–	–	13
	Bit Set	–	–	–	13
	Compara maschera (WORD)	–	–	–	25
	Compara maschera (DWORD)	–	–	–	25
Sposta dati	Sposta (INT)	–	–	–	10
	Sposta (BIT)	–	–	–	13
	Sposta (WORD)	–	–	–	10
	Sposta (REALE)	–	–	–	13
	Sposta blocco (INT)	–	–	–	28
	Sposta blocco (WORD)	–	–	–	28
	Sposta blocco (REALE)	–	–	–	13
	Cancella blocco	–	–	–	11
	Solleva registro (BIT)	–	–	–	16
	Solleva registro (WORD)	–	–	–	16
	Sequenziatore Bit	–	–	–	16

Tabella Funzioni

Gruppo	Funzione	Abilitata	Disabilitata	Incrementi	Quantità
Tabella	Sposta Array				
	INT	–	–	–	22
	DINT	–	–	–	22
	BIT	–	–	–	22
	BYTE	–	–	–	22
	WORD	–	–	–	22
	Cerca uguale				
	INT	–	–	–	19
	DINT	–	–	–	22
	BYTE	–	–	–	19
	WORD	–	–	–	19
	Cerca non uguale				
	INT	–	–	–	19
	DINT	–	–	–	22
	BYTE	–	–	–	19
	WORD	–	–	–	19
	Cerca maggiore di				
	INT	–	–	–	19
	DINT	–	–	–	22
	BYTE	–	–	–	19
	WORD	–	–	–	19
	Cerca maggiore di/uguale				
	INT	–	–	–	19
	DINT	–	–	–	22
	BYTE	–	–	–	19
	WORD	–	–	–	19
	Cerca mano di				
	INT	–	–	–	19
	DINT	–	–	–	22
	BYTE	–	–	–	19
	WORD	–	–	–	19
	Cerca meno di/uguale				
	INT	–	–	–	19
	DINT	–	–	–	22
	BYTE	–	–	–	19
	WORD	–	–	–	19

funzioni di conversione e controllo

Gruppo	Funzione	Abilitata	Disabilitata	Incrementi	Quantità
Conversione	Converti in INT	–	–	–	–
	Converti in BCD4	–	–	–	–
	Converti in REALE	–	–	–	–
	Converti in WORD	–	–	–	–
	Tronca a INT	–	–	–	–
	Tronca a DINT	–	–	–	–
Controllo	Richiama una subroutine	–	–	–	–
	Esegui I/O	–	–	–	–
	Algoritmo PID – ISA	–	–	–	–
	Algoritmo PID – IND	–	–	–	–
	Fine istruzione	–	–	–	–
	Richiesta di assistenza				
	#6	–	–	–	–
	#7 (Leggi)	–	–	–	–
	#7 (Imposta)	–	–	–	–
	#14	–	–	–	–
	#15	–	–	–	–
	#16	–	–	–	–
	#18	–	–	–	–
	#23	–	–	–	–
	#26//30	–	–	–	–
	#29	–	–	–	–
	MCR/ENDMCR Annidati	–	–	–	–
	Combinato	–	–		
	COMM_REQ	–	–	–	–

G

I

L

M

N

O

P

R

U

W